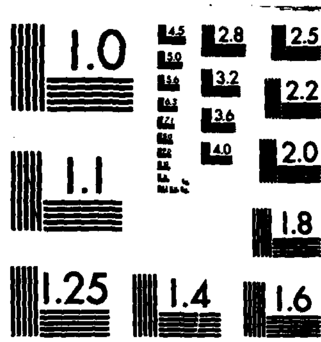
AD-A154 460 APPLICATION OF FLOQUET THEORY TO HELICOPTER BLADE FLAPPING STABILITY(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING J K MARCH DEC 84 AFIT/GAE/AA/84D-13 1/3

UNCLASSIFIED F/G 1/3 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

1

AD-A154 460

APPLICATION OF FLOQUET THEORY TO
HELICOPTER BLADE FLAPPING STABILITY

THESIS

James K. March
Captain, USAF

AFIT/GAE/AA/84D-13

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

85 5 07 171

AFIT/GAE/AA/84D-13

1

APPLICATION OF FLOQUET THEORY TO HELICOPTER BLADE FLAPPING STABILITY

THESIS

James K. March
Captain, USAF

AFIT/GAE/AA/84D-13

Approved for public release; distribution unlimited

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GAE/AA/84D-13	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: School of Engineering; 6b. OFFICE SYMBOL (If applicable): AFIT/EN	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State and ZIP Code): Air Force Institute of Technology, Wright-Patterson AFB, Ohio 45433	7b. ADDRESS (City, State and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION; 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS.: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO.
11. TITLE (Include Security Classification): See Box 19	
12. PERSONAL AUTHOR(S): James K. March, B.S., Capt, USAF	

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
MS Thesis	FROM ____ TO ____	1984 December	208

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	Helicopter Rotors, Rotor Blades, Fourier Series, Numerical Integration, Control Theory
01	03		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: APPLICATION OF FLOQUET THEORY TO HELICOPTER BLADE FLAPPING STABILITY

Thesis Chairman: Dr. Robert A. Calico

Approved for public release: IAW AFR 190-17.
LYNN E. WOLAVER 17 Apr 85
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
Dr. Robert A. Calico	513-255-2362	AFIT/ENY

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

The purpose of this investigation was to explore the flapping stability of a helicopter blade in forward flight. The equations of motion for the flapping motion of the blade were converted from nonlinear differential equations with periodic coefficients to linear periodic differential equations through the assumption of a rigid blade where the elastic flapping deflections are negligible as compared to the rigid body flapping rotations about the flapping hinge. Aeroelastic effects were not considered.

The stability of the homogeneous part of the flapping motion linearized periodic differential equations was examine through the application of Floquet theory. The flapping blade motion was simulated over one period to derive the elements of the monodromy matrix. The monodromy matrix was next transformed into Jordan normal form through a similarity transformation to obtain its characteristic values and eigenvectors. The characteristic values were converted to their respected Poincare' exponents and the periodic eigenvectors composition was determined and transformed into Fourier series representations.

A feedback controller was constructed using Floquet theory for the unstable blade flapping motion which the blade was unrestrained, had an advance ratio of 2.4, a lock number of 8, and a tip loss factor of 0.97. The performance of the feedback controller was then compared to the unstable case.

AFIT/GAE/AA/84D-13

APPLICATION OF FLOQUET THEORY TO HELICOPTER BLADE FLAPPING STABILITY

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Aeronautical Engineering

James K. March, B.S.
Captain, USAF

December 1984

Approved for public release; distribution unlimited

Preface

The purpose of this study was to conduct an analytical experiment in the application of Floquet theory to the helicopter rotor blade flapping motion problem. Prior to this time, Floquet application had been restricted to examining the stability of the blade flapping motion. This study extends the application to using Floquet theory as a tool for blade feedback control design synthesis.

In conducting this study, several computer algorithms were used to derive the key information from the blade equations of motion needed to design the feedback controller. All the computer software needed to conduct this study is included in the appendices of the text in their order of implimentation. This software is written in Texas Instruments Extended Basic, but is easily convertible to another computer language. Output from the algorithms is also included in the appendices for model comparisons.

In conducting this analysis, I am deeply indebted to others for their encouragement and assistance. First, a word of thanks to my thesis advisor, Dr. Robert Calico, and to Dr. William Wiesel for the countless hours of discussions and insights received on the idiosyncrasies of Floquet theory which allowed me to apply it to this study. And, a sincere thanks to my wife Jane and daughters Elizabeth and Victoria for their understanding and unending support through this sometimes trying odyssey.

James K. March

Table of Contents

List of Figures

List of Tables

List of Symbols

a	lift coefficient versus angle of attack curve slope
a_k, b_k	Fourier series coefficients
B	tip loss factor
$C(\psi)$	aerodynamic damper characteristic of blade on flap moment
c	chord of airfoil section
C_l	lift coefficient for a given blade element angle of attack
dC	differential centrifugal force
$f(\psi)$	forcing function due to aerodynamic forces on blade
$f(t)$	Fourier series
g	gravitational acceleration constant
$[G(t)]$	mode controllability matrix
I	blade moment of inertia about the Z_{hinge} axis
$K(\psi)$	aerodynamic spring characteristic on flap moment
$[K(t)]$	gain matrix
K_o	flap restraint spring constant
L	lift generated by blade
m	element mass of blade element
P^2	blade natural frequency in hover
R	total length of blade
r	distance along blade centerline measured from hinge
T	period of the periodic system

u_P	velocity component in Z_{hinge} direction
u_R	radial velocity component in X_{blade} direction
u_T	velocity component in Z_{hinge} direction
U	forward velocity vector of helicopter
x	nondimensional distance ratio; r/R
X_{hinge}, Y_{hinge}, Z_{hinge}	coordinate reference frame of hinge
X_{blade}, Y_{blade}, Z_{blade}	coordinate reference frame of blade
α	blade element angle of attack
β	blade flap angle
$\dot{\beta}$, $\ddot{\beta}$	1st and 2nd derivative of blade flap angle with respect to time
β', β''	1st and 2nd derivative of blade flap angle with respect to blade azimuth angle
γ	blade lock number
ϵ	angular width of mixed flow region across the rotor plane
η	modal variable
θ	blade element pitch angle to horizontal X_{hinge}, Y_{hinge} plane
θ_o	pitch angle of blade root
θ_1	built in linear twist angle along blade's longitudinal axis
θ_s, θ_c	cyclic pitch control terms
λ	inflow ratio
λ_i	characteristic value

μ	blade advance ratio
ρ	mass density in free stream
Φ	state transition matrix
ϕ	velocity vector angle to $\dot{X}_{hinge}$, Y_{hinge} plane
ψ	blade azimuth angle measured from tail of helicopter in counter-clockwise direction
Ω	rotor angular velocity
ω_i	Poincare' exponent

AFIT/GAE/AA/84D-13

Abstract

The purpose of this thesis ~~investigation~~ was to explore the flapping stability of a helicopter rotor blade in forward flight. The equations of motion for the flapping motion of the blade were converted from nonlinear differential equations with periodic coefficients to linear periodic differential equations through the assumption of a rigid blade where the elastic flapping deflections are negligible as compared to the rigid body flapping rotations about the flapping hinge. Aeroelastic effects were not considered.

The stability of the homogeneous part of the flapping motion linearized periodic differential equations was examine through the application of Floquet theory. The flapping blade motion was simulated over one period to derive the elements of the monodromy matrix. The monodromy matrix was next transformed into Jordan normal form through a similarity transformation to obtain its characteristic values and eigenvectors. The characteristic values were converted to their respected Poincare' exponents and the periodic eigenvectors composition was determined and transformed into Fourier series representations.

A feedback controller was constructed using Floquet theory for the unstable blade flapping motion case, where; P^2=1.0, μ =2.4, γ =8, and B=0.97. The performance of the feedback controller was then compared to the unstable case.

Key words — Fourier series, Numerical integration, Control theory

APPLICATION OF FLOQUET THEORY TO HELICOPTER BLADE FLAPPING STABILITY

I. Introduction

The helicopter rotor blade may be modelled as a spring mass damper system with periodic changing spring and damper characteristics. The origin of these characteristics are due to the elastic blade exposed to varying aerodynamic forces as well as aeroelastic effects. In forward flight, these effects are more pronounced as the forward speed of the helicopter increases due to the variable flow region across the rotor plane.

The first successful helicopters were made possible largely by the development of the fully articulated rotor hub, where blade motion was accommodated by flapping and lead-lag hinges at the blade root to reduce the bending stresses at the root (33). This technique is similar in principle to increasing the thickness of a propeller blade for structural considerations (43:285).

The root blade itself can be considered as an elastic beam. The resulting equations of motion of the blade are nonlinear differential equations with periodic coefficients. By assuming small blade pitch and twist along with accounting for the nonlinear terms in the inertial and aerodynamic forces, these equations can be transformed into linear time-periodic differential equations. With the further assumption of low

forward velocity, they can be treated as linear constant coefficient differential equations.

Initially, the differential equations were assumed to have constant coefficients due to the low velocity of earlier helicopter designs. As the forward speed increased, the periodicity became more apparent. The first attempt to handle this problem converted the blade equations of motion to the form of the generalized Hill equation (21). This approach permitted the stability determination of the blade motion through the properties of the solution of the Hill equation.

Through the introduction of the digital computer, stability determination was conducted by simulating the motion of the blade over several revolutions. However, with the increase in complexity of the mathematical model due to the incorporation of detailed aeroelastic effects on the blade, the efficient application of Floquet theory was discovered. Since that time, Floquet theory has been the standard technique to parametrically study the stability boundaries of helicopter blade motion for various designs and flight conditions.

The purpose of this research is to investigate the flapping stability of a helicopter blade in forward flight. The first step in this investigation was to determine the blade flapping equation of motion. The assumptions made in this development were that the blade was rigid with no twist and that the flap and pitch angles were small. The cross sectional center of mass was also assumed to coincide with the elastic axis of

the blade. Thus, the only momentum terms that needed to be considered in the development were those due to inertial forces, aerodynamic forces, and flapping restraint contributions. Other assumptions made in this development are addressed further in the text of this study.

The next objective of this research was to explore the applicability of Floquet theory to this problem. The most obvious application of this theory was to investigate the stability of the rotor blade for different flight and design conditions. However, the use of Floquet theory was carried one step further by experimenting with its application to feedback control synthesis. This technique has been applied to other control design problems dealing with periodic systems (3,4,44).

One comment which is made is that Floquet theory predicts the properties and characteristics of the solution, but it does not generate the solution itself. To obtain the solution to the Floquet problem, numerical methods are required. The numerical methods used in this study will also be presented.

To understand the impact of Floquet theory on feedback controller design, it is worth noting the past basic design steps which have been used. The initial blade controllers consisted of a controllable pitch anti-torque rotor for control about the vertical axis, and a cyclic pitch control for the main rotor blades for control of the other two axes (45:63). The gains associated with these type of controllers were determined by first assuming low forward speed. Through the use of

classical and modern control theory, gain selection was made (16,18,30,34). Once the desired control properties were obtained for this time invarient case, the forward speed of the helicopter was increased and the articulated rotor was digitally simulated for several conditions to establish stability boundaries. The final gains for these controllers were thus determined by varying the gains until the poles of the dynamic region lie within the predicted stable region of response.

Another approach taken was to identify the components of blade motion, then build a controller to control these components using the same approach as previously discussed (1,20, 35,36). Floquet theory was then applied to parametrically study the stability boundaries of the system.

The introduction of Floquet theory will change this design process by giving the designer a direct access to pole location control for the periodic system. Rather than beginning at the time invarient condition of the blade flapping equation of motion, the designer will be able to start from the time periodic condition. This study will show why this can be done and will give examples of the control design process.

Because this is a blade motion stability analysis, a detailed aeroelastic effects study will not be necessary. However, it should be mentioned that the techniques discussed in this study can be equally applied in the aeroelastic stability problem (5,10,12,13).

II. Helicopter Rotor Flapping Dynamics

As described in the previous section, the helicopter rotor blade can be modelled as a spring mass damper system with periodic changing spring and damper characteristics. These characteristics are excited by the aerodynamic forces due to inflow about the rotor blade, blade twist, cyclic pitch control, and collective pitch control (41). Thus, the blades are hinged so as to prevent the transmission of the alternating moments of the lateral aerodynamic forces on the blade in forward flight onto the rotor shaft (21,28:150). The derivation of the flapping dynamic equation of motion requires the knowledge of these aerodynamic forces and moments, along with the elastic and inertial properties of the blade.

The helicopter rotor can be considered as being composed of elastic blades. The exact description of the flapping dynamics requires a detailed consideration of the elastic flexibility of each blade (38). There is, however, a wide array of assumptions that can be made which lead to a variety of models, starting with the simple and computationally efficient model to those which are capable of simulating unsteady flow in high detail (7). The decision as to what assumptions to make and what model to use depends on the purpose of the model and the penalty one is willing to pay for more sophistication (38).

Keeping in mind that the purpose of this study is to examine the blade flapping stability and control in forward

flight, it is justifiable to assume a rigid blade where the elastic flapping deflections are negligible compared to the rigid body flapping rotations about the flapping hinge (38). It has been shown (8,14,33,38) that this approach can also be used for hingeless rotors. This is done by specifying a rigid blade with a selected offset hinge and an appropriate spring constant to match the properties of the hingeless rotor.

The purpose of this section is to present the derivation of the dynamic flapping equation of motion of the blade. This will be accomplished using blade element theory to represent the inertial, aerodynamic, and restoring moments on the blade flapping motion. Through a coordinate transformation, the equation will be converted to a usable, state space representation which was modelled to conduct the stability and control investigation of this study.

It is interesting to note here why blade element theory was employed in the development of these equations. Blade element theory has been extensively used and has strong foundations in propeller theory (43:326). It allows for the convenient use of experimentally derived airfoil data for predicting blade performance to within 10 percent accuracy, depending on the quality of airfoil data used.

Coordinate Frame

The coordinate reference frame used in this derivation can be found in Figure 1. The X_{hinge} and Y_{hinge} axes define the horizontal plane of the hinge. Z_{hinge} is defined as being

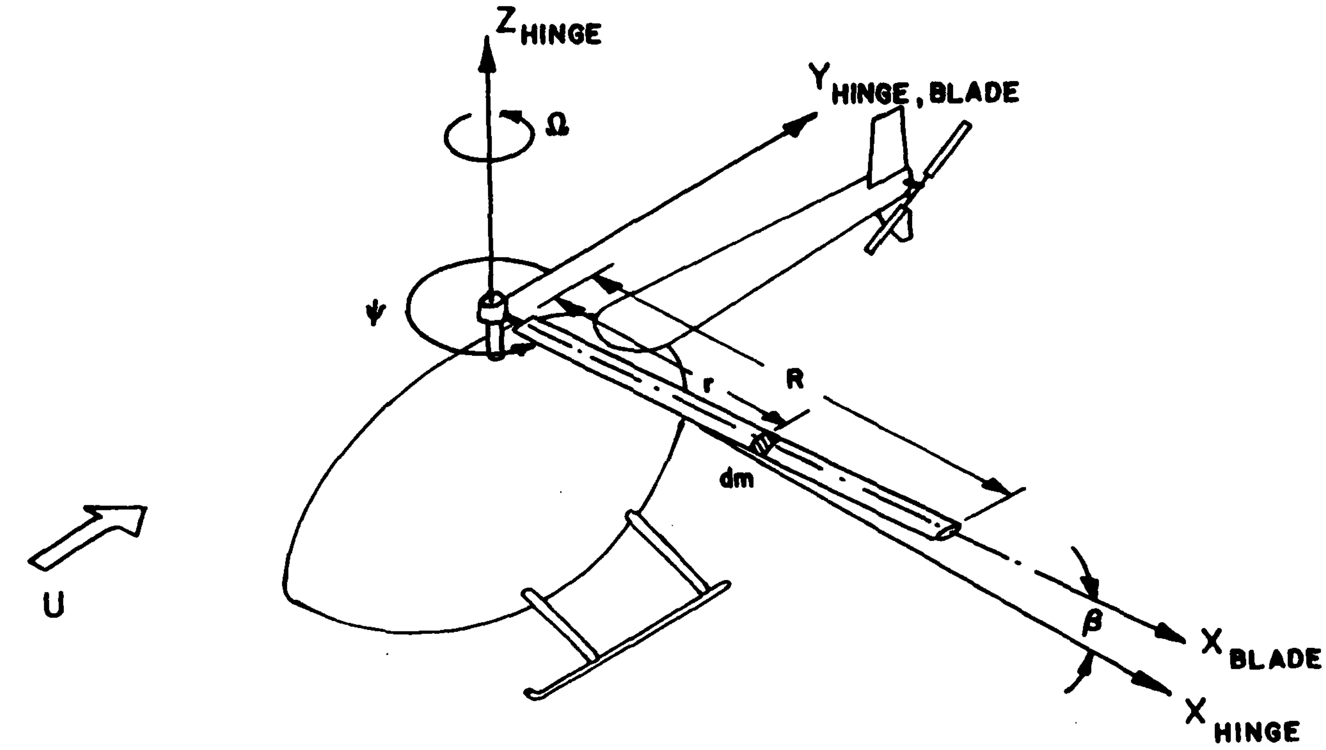


Figure 1. Blade Coordinate Frame

perpendicular to the X_{hinge} and Y_{hinge} axes and is positive upwards.

The X_{blade} axis is defined along the centerline of the blade. The offset angle between the X_{blade} and X_{hinge} axes define the flap angle β . The Y_{blade} axis corresponds to the Y_{hinge} axis, while the Z_{blade} axis is offset by the angle β from the Z_{hinge} axis.

The blade rotates about the Z_{hinge} axis at the constant angular velocity Ω . The azimuth angle ψ is measured from the tail of the helicopter in the positive counterclockwise direction toward the nose.

The helicopter is assumed to be moving with a constant forward velocity, V.

The blade flapping equation of motion will be derived for a blade element dm which is located at the distance r measured from the blade hinge. The overall length of the blade will be refered to as R.

Flapping Hinge Moments

Using the assumption of small helicopter roll, pitch, and yaw rates compared to the angular velocity Ω of the rotor, along with a negligible offset of the blade hinge from the rotor hub center, the blade hinge can be treated as an inertially fixed point in space. These assumptions allow for the sum of the moments about the hinge to equal the time rate of change of its moment of momentum. The sum of the moments will be taken in the Z_{hinge} direction.

The external forces on a blade element are depicted in Figure 2 and consist of the element lift force, weight, and centrifugal force generated by the angular velocity of the blade. Aerodynamic drag on the element is not included since it does not contribute to the flapping hinge moment.

Using the assumption of a small angle, the inertial force about the flapping hinge moment is;

$$m\ddot{Z}_{hinge} = mr\ddot{\beta} \qquad (1)$$

which acts at a distance r from the hinge. The centifugal force on the blade element is;

$$mr\,\Omega^2 \qquad (2)$$

This force component acts with a moment arm, $r\beta$, on the hinge. Gravitational force of magnitude, g dm, acts at a distance r from the hinge.

The differential thrust generated by the blade element is expressed as;

$$dT = Ldr \qquad (3)$$

where L represents the lift generated by the element. This force acts with a moment arm, r, on the blade hinge.

Summing the external force moments to the time rate of change of the flap hinge's moment of momentum (Equation (1)),

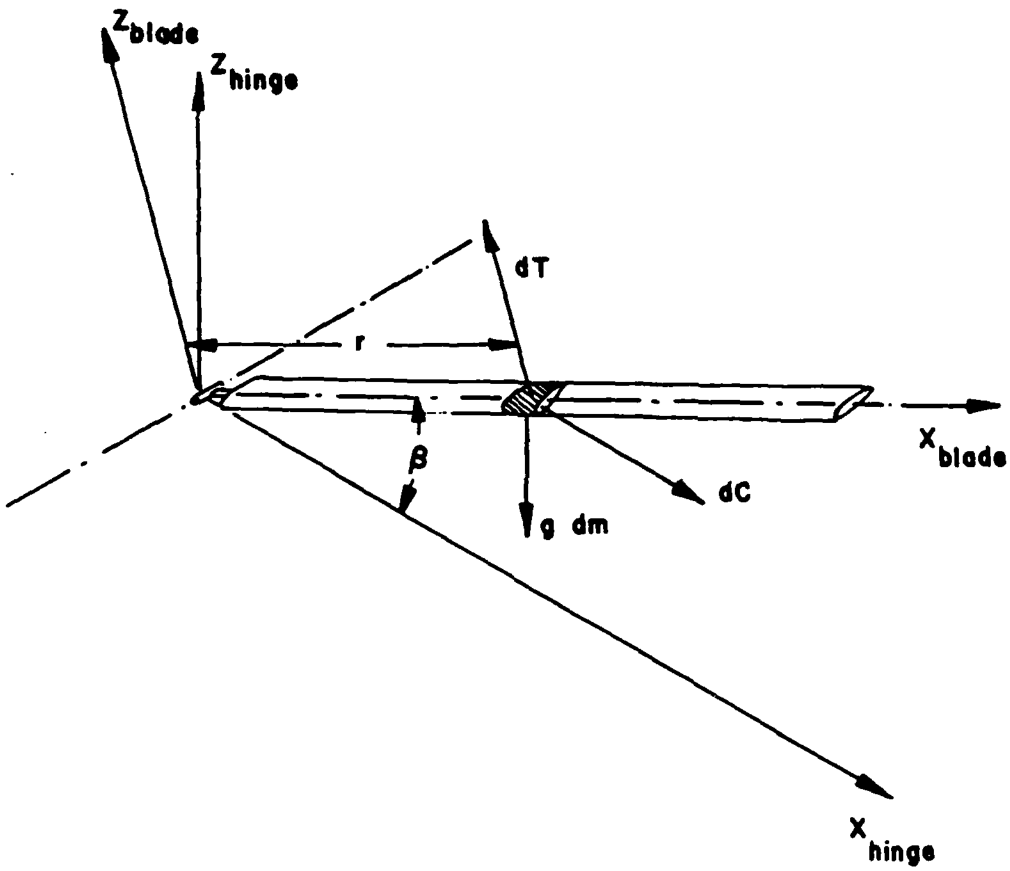


Figure 2. Blade External Forces

yields the expression;

$$\int_0^R mr\ddot{\beta}\, r\,dr + \int_0^R m\,\Omega^2 r(r\beta)\,dr + \int_0^R r\,dT + \int_0^R rg\,dm = 0 \qquad (4)$$

Recognizing that the blade moment of inertia about the Z_{hinge} axis as;

$$I = \int_0^R mr^2 dr \qquad (5)$$

Equation (4) can be rewritten as;

$$\ddot{\beta} + \Omega^2\dot{\beta} = (1/I)\left\{\int_0^R r\,dT - \int_0^R rg\,dm\right\} \qquad (6)$$

This is the initial form of the blade flap angle equation of motion. Further refinements are necessary to make Equation (6) more useful.

<u>Blade Aerodynamics</u>

The aerodynamic force on the blade element will next be examined using the quantities as depicted in Figure 3.

The forward velocity on the element is depicted as U. This velocity has a component in the Z_{hinge} direction u_p. It also has a radial component, u_R, in the X_{blade} direction.

The blade is at an angle of attack, α, to the velocity vector. This angle is the sum of the blade pitch angle to the horizontal X_{hinge}, Y_{hinge} plane, θ, and the velocity vector angle to the X_{hinge}, Y_{hinge} plane, ϕ. It is mathematically;

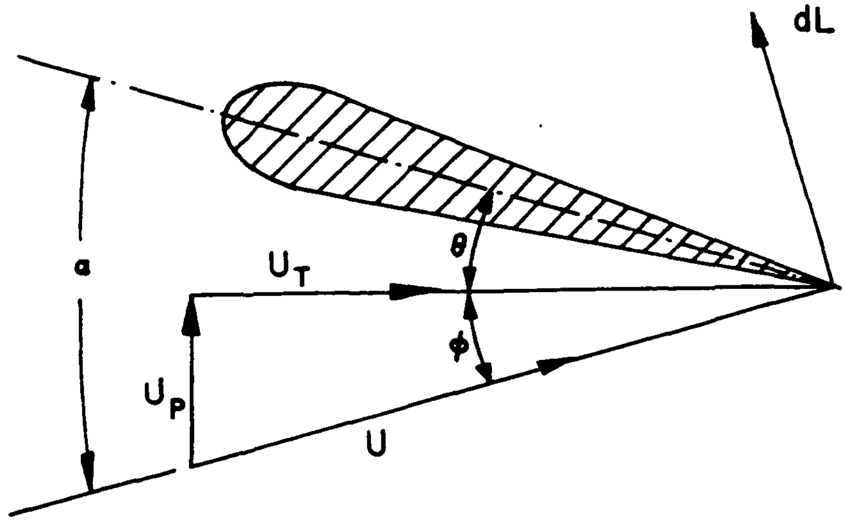


Figure 3. Blade Aerodynamics

$$\alpha = \theta + \phi \tag{7}$$

Assuming small ϕ, the following trigonometric relationships can be applied;

$$\sin\phi = \tan\phi = u_P/u_T \tag{8}$$

where u_T represents the tangential velocity to the blade element. Using Equation (8), Equation (7) is rewritten as;

$$\alpha = \theta + u_P/u_T \tag{9}$$

The lift equation for the blade element is written as;

$$L = \tfrac{1}{2}\rho U^2 c C_l \tag{10}$$

where c represents the chord of the airfoil section and C_l represents the lift coefficient for a given α . C_l can be approximated by;

$$C_l = a\,\alpha \tag{11}$$

where a represents the C_l versus α curve slope. Using Equation (9), Equation (11) is written as;

$$C_l = a\left\{\theta + u_P/u_T\right\} \qquad (12)$$

By recognizing that for small ϕ , $U=u_T$ and the application of Equation (12), Equation (10) is transformed into;

$$L = \tfrac{1}{2}\rho\, u_T^2 ca\left\{\theta + u_P/u_T\right\} \qquad (13)$$

The differential thrust, as expressed in Equation (3), using Equation (13), is now written as;

$$dT = \tfrac{1}{2}\rho\, u_T^2 ca\left\{\theta + u_P/u_T\right\} dr \qquad (14)$$

The terms, u_T, u_P, and of Equation (14) will next be further refined.

As previously mentioned, u_T represents the velocity in the Y_{hinge} direction, u_P represents the velocity component in the Z_{hinge} direction, and u_R represents the velocity in the X_{blade} direction.

The term, u_T, is composed of two forward velocity components of the blade through a stationary air mass. These components of u_T are the tangent velocity of the blade element, $r\Omega$, due to the angular rotation of the blade, and the tangent component of the helicopter's forward velocity to the blade, $V\sin\psi$. Thus, u_T can be mathematically expressed as;

$$u_T = r\Omega + V\sin\psi \qquad (15)$$

The radial velocity component, u_R, along the X_{blade} axis is composed of the normal velocity component of forward helicopter motion and is expressed as;

$$u_R = V\cos\psi \tag{16}$$

The perpendicular velocity component of the blade, u_P, is composed of the velocity component in the Z_{hinge} direction due to the flap angle velocity, $\dot{\beta}$, and the induced velocity due to downwash, v, as predicted by finite airfoil theory. By involking the assumption of β being small, u_P can be expressed as;

$$u_P = v - \beta V\cos\psi - r\dot{\beta} \tag{17}$$

Nondimensionalizing the velocity components by the term, $R\Omega$, Equations (15)-(17) can be written as;

$$U_T = x + \mu\sin\psi \tag{18a}$$

$$U_R = \mu\cos\psi \tag{18b}$$

$$U_P = \lambda - \mu\beta\cos\psi - x\dot{\beta}/\Omega \tag{18c}$$

where λ represents the inflow ratio, μ is defined as the blade advance ratio, and x represents the nondimensional distance along the blade. The terms, μ and x are represented by;

$$\mu = V/(R\Omega) \tag{19}$$

$$x = r/R \tag{20}$$

Nondimensionalizing Equation (14) by the terms, $(\frac{1}{2}\rho ca)$ and R, along with substituting the nondimensional velocity components of Equations (18a)-(18c), yields the nondimensional aerodynamic moment for the blade;

$$\left[\tfrac{1}{2}\rho ca\,\Omega^2 R^4\right]\int U_T^2 x(U_P/U_T + \theta\ )dx \tag{21}$$

For this study, the local pitch of the blade was considered to be composed of four components. The first component, θ_o, represents the pitch angle of the blade root. The second component, θ_1, represents a built in linear twist angle along the blade's longitudinal axis. The third and fourth components, θ_s and θ_c, represent the cyclic pitch control terms. These control terms are a function of blade azimuth angle and represent the change in blade pitch necessary for directional control of the thrust vector across the rotor plane for helicopter horizontal and vertical control.

The pitch angle, θ, can thus be represented as;

$$\theta = \theta_o + x\theta_1 + \theta_s \sin\psi + \theta_c \cos\psi \tag{22}$$

Substitution Equations (22) and (18c) into Equation (21)

$$\lambda \int U_T \, x \, dx + \theta_o \int U_T^2 \, x \, dx + \theta_1 \int U_T^2 \, x^2 dx + \theta_s \int U_T^2 \, x \sin\psi \, dx$$

$$+ \theta_c \int U_T^2 \, x \cos\psi \, dx - \beta \int U_T \, x \mu \cos\psi \, dx \tag{23}$$

$$- (\dot{\beta} / \Omega) \int U_T \, x^2 dx$$

To establish the integration boundaries of Equation (23), two concepts must be addressed.

The first concept is the blade tip loss factor, B. Due to the limitations of blade element theory in predicting the performance of a blade of finite length, adjustments must be made to compensate for the loss of lift generating ability at the blade tip. This adjustment is made by defining the blade tip loss factor as a limit of integration along the nondimensional blade longitudinal direction. The ad hoc selected numerical value for B is 0.97, which gives good correlation with experimental data (28:60).

The second concept which must be addressed is the definition of flow regions across the rotor plane (28:152,41,8). These regions, as depicted in Figure 4, are refered to as region 1 for normal flow, region 2 for mixed flow, and region 3 for reversed flow (38).

The first region is considered normal flow because the air approaches the blade element from the leading edge. This region is defined from where the azimuth angle equals 0 degrees to where it equals 180 degrees. The third region is called reversed flow because the air approaches the blade from the trailing edge. The angular boundaries for this region are defined as;

$$\psi = \pi + \epsilon \text{ to } \psi = 2\pi - \epsilon \qquad (24)$$

where ϵ is defined as;

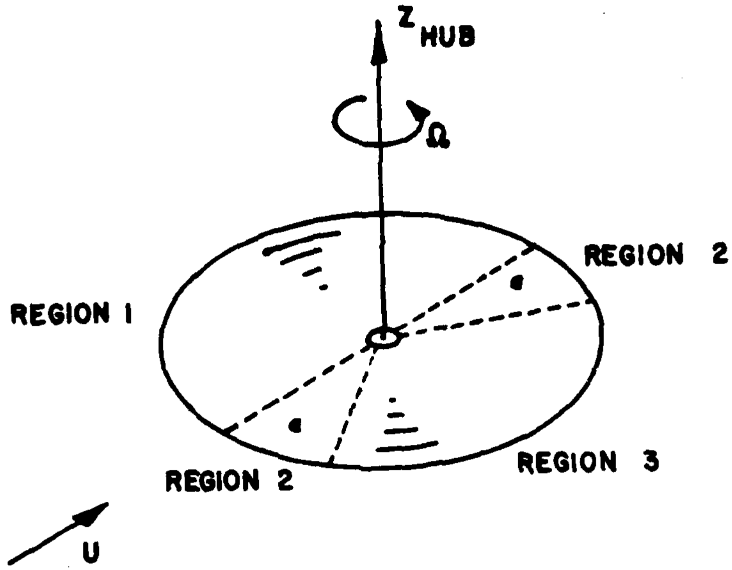


Figure 4. Flow Regions

$$\sin\epsilon = B/\mu \tag{25}$$

In the reversed flow region, the lift coefficient is assumed to be the negative of the lift coefficient for the given angle of attack in normal flow (41).

The second region of flow across the rotor is called mixed flow because it represents a combination of normal flow across the tip, and reverse flow close to the hub. The angular regions that define its boundaries are;

$$\psi = \pi \text{ to } \pi + \epsilon \text{ and } \psi = 2\pi - \epsilon \text{ to } 2\pi \tag{26}$$

To handle this condition, the appropriate aerodynamic expressions must be evaluated for each region. This means that the rotor flow model must be an integral part of the blade flapping simulation model. There have been alternative approaches to this condition (11) which gives approximately the same azimuth angle boundaries for the different flow conditions.

Other flow effects which can be considered, but are out of the scope in this study, are those due to transonic effects (23), wake effects (25,26), and atmospheric turbulence (15,37).

For the mixed flow region, the normal velocity approaching the trailing edge of the blade is equal to the normal velocity approaching the leading edge of the blade at the non-dimensional blade location, $-\mu\sin\psi$. This establishes the boundary for normal and reversed flow on the blade when it is

in the mixed flow region.

Using the definitions of tip loss factor along with the three distinct flow regions, the nondimensional aerodynamic moment equation has for its integration boundaries;

Region 1: $\int_{0}^{B}$

Region 2: $\int_{-\mu \sin\psi}^{B} \quad - \int_{0}^{-\mu \sin\psi}$

Region 3: $-\int_{0}^{B}$

Using the integration boundaries along with combining terms in β and $\dot{\beta}$, Equation (23) can be expressed as;

$$\left[\frac{\Omega^2 R^4 \rho\, ca}{2}\right] \left\{ \sum f(\psi) - \beta K(\psi) - (\dot{\beta}/\Omega) C(\psi) \right\} \tag{27}$$

where

$$K(\psi) = \mu \int U_T x \cos\psi \, dx \tag{28}$$

$$C(\psi) = \int U_T x^2 dx \tag{29}$$

Equation (28) represents the aerodynamic spring characteristics on the flap moment. Equation (29) represents the aerodynamic damper characteristics of the blade on the flap moment. The term, $f(\psi)$, represents a forcing function due to aerodynamic forces resulting from applied pitch controls, initial

blade pitch, built in linear twist, and inflow effects. Thus, $f(\psi)$ represents all the aerodynamic and control effects built around the flapping dynamics of the blade.

The expression for $K(\psi)$ and $C(\psi)$ have been determined for each flow region (33,36,21), and are presented in Table I.

Substituting Equation (27) into Equation (6) yields the flapping equation of motion;

$$\ddot{\beta} + \Omega^2\beta = \left[\frac{\Omega^2 R^4 \rho\, ca}{2I}\right]\left\{\sum f(\psi) - \beta K(\psi) - (\dot{\beta}/\Omega)C(\psi)\right\} + \int_0^R rg\,dm \qquad (30)$$

Defining the blade lock number as;

$$\gamma = \frac{R^4 \rho\, ca}{I} \qquad (31)$$

and introducing a flap restraint spring with the spring constant K_o, which produces a moment at the flap hinge of;

$$K_o \beta \qquad (32)$$

to counteract the aerodynamic moment at the flap hinge, Equation (30) can be written as;

$$\ddot{\beta} + \frac{\gamma\Omega}{2} C(\psi)\dot{\beta} + \frac{\gamma\Omega^2}{2}\left[\frac{2}{\gamma}P^2 + K(\psi)\right]\beta = \frac{\gamma\Omega^2}{2}\sum f(\psi) + \int_0^R rg\,dm \qquad (33)$$

MOMENT FUNCTIONS

$K(\psi)$	$K_1 = \frac{1}{3} B^3 \mu \cos\psi + \frac{1}{4} B^2 \mu^2 \sin 2\psi$	NORMAL FLOW
	$K_2 = K_1 + \mu^4 (- \frac{1}{12} \sin 2\psi + \frac{1}{24} \sin 4\psi)$	MIXED FLOW
	$K_3 = -K_1$	REVERSED FLOW
$C(\psi)$	$C_1 = \frac{1}{4} B^4 + \frac{1}{3} B^3 \mu \sin\psi$	NORMAL FLOW
	$C_2 = C_1 + \mu^4 (\frac{1}{16} - \frac{1}{12} \cos 2\psi + \frac{1}{48} \cos 4\psi)$	MIXED FLOW
	$C_3 = -C_1$	REVERSED FLOW

TABLE I

where;

$$P^2 = 1 + \frac{K_o}{\Omega^2 I} \tag{34}$$

Equation (34) represents the natural frequency of the flapping motion when the helicopter is hovering. Equation (31) represents the ratio of the aerodynamic forces to inertial forces.

By choosing an appropriate hinge spring constant K_o, a hingeless rotor blade can also be examined through Equation (33).

For this stability study, only the homogeneous part of Equation (33) was considered. The only situation where the forcing functions of the heterogeneous part of Equation (33) would impact stability, would be when the homogeneous part was unstable (21). For a stable system, the effects of the forcing function terms would be a different equilibrium pitch and flap angle position other than zero at some azimuth angle.

Stability Model

To eliminate the dependence of rotor angular velocity, Ω, on the solution, a coordinate transformation was made. Assuming a constant rotational velocity and using the chain rule, $\dot{\beta}$ and $\ddot{\beta}$ were transformed to be a function of azimuth angle rather than time. This transformation took the form;

$$\frac{d}{dt}\beta = \frac{d\beta}{d\psi}\frac{d\psi}{dt} = \Omega\frac{d\beta}{d\psi} = \Omega\beta' \tag{35}$$

and;

$$\frac{d}{dt}\left[\frac{d\beta}{dt}\right] = \Omega^2 \frac{d^2\beta}{d\psi^2} = \Omega^2 \beta'' \tag{36}$$

The new form of the flapping equation after making the substitution of Equations (35) and (36), is;

$$\beta'' + \frac{\gamma}{2} C(\psi)\beta' + \frac{\gamma}{2}\left[\frac{2}{\gamma} P^2 + K(\psi)\right]\beta = 0 \tag{37}$$

Converting Equation (37) into state space representation yields the expression;

$$\begin{bmatrix} \beta' \\ \beta'' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{\gamma}{2}\left[\frac{2}{\gamma} P^2 + K(\psi)\right] & -\frac{\gamma}{2} C(\psi) \end{bmatrix} \begin{bmatrix} \beta \\ \beta' \end{bmatrix} \tag{38}$$

Equation (38) was used as the simulation model for the blade flapping motion. The rate of change of flap angle, β', was initially set at zero, with the flap angle perturbed from the equilibrium position at the zero azimuth degree location.

Had the matrix coefficients of Equation (38) been constant values, standard classical and modern control procedures, such as root locus and Bode analyses, could have been used for stability determination and control design. However, with the periodic coefficients, these techniques could not be used and a different approach had to be taken. This approach centers around Floquet theory which will be addressed in the next section.

III. Floquet Theory

From Section II, it was demonstrated that the helicopter blade flapping equation of motion represented a nonlinear, time-periodic differential equation. It was further demonstrated that, through the use of appropriate assumptions and the physical characteristics of the blade, these equations can be transformed into linear, time-periodic differential equations such as Equation (38). For low speed forward flight where the advance ratio is relatively low, it can be assumed that the coefficients of the flapping equation of motion are constant values. However, for the faster forward velocity case, the periodic nature of the differential equation returns.

The most straightforward method of dealing with the stability of this type of periodic system is through the application of Floquet theory (9,21,36). The major utility of Floquet theory is to determine the stability of the dynamic system. For the helicopter blade application, the theory has been used only for this purpose and not as a design synthesis tool for feedback control.

The purpose of this section is to lay out the general foundations of Floquet theory. The consequences of the theory will be addressed along with some insights on how the theory can be applied for feedback control synthesis.

Foundations of Floquet Theory

To begin the development of the basic ideas of Floquet theory, a periodic system will be defined as;

$$[\dot{X}] = A(t)\,[X] \tag{39}$$

where the matrix $A(t)$ is periodic over some time interval T. The solution to this differential equation is;

$$X(t) = \Phi(t,0)\,X(0) \tag{40}$$

where $\Phi(t,0)$ represents the state transition matrix. The solution to the state transition matrix is obtained by solving the differential equation;

$$\frac{d}{dt}\,\Phi(t,0) = A(t)\,\Phi(t,0) \tag{41}$$

which has the identity matrix for its initial condition.

Floquet theory states (4,9) that the state transition matrix can be written as;

$$\Phi(t,0) = F(t)\,\exp\left\{[J]\,t\right\} F^{-1}(0) \tag{42}$$

where t is any time greater than zero. In Equation (42), the matrix $[J]$ represents a constant matrix which, in most cases, is constructed in Jordan normal form. The diagonal entries of $[J]$ are refered to as the Poincare' exponents, which are similar to the eigenvalues of a constant coefficient differential equation. The matrix $F(t)$ is periodic with the same period of the dynamic system.

It should be noted here that Floquet theory does not yield the solution to the problem, but gives information about the form and properties of the solution (9). The knowledge over one period of time determines the solution to the homogeneous system of Equation (39) through Equation (42). Thus, solving the Floquet problem for all time requires finding the constant matrix $\left[J\right]$ and the periodic matrix F(t) over one period.

Since F(t) is periodic, the elements of F(t) at the beginning of the period equals the elements at the termination of the period. Using this information to set the initial conditions, solving Equation (42) yields the solution;

$$\Phi(T,0) = F(0)\ \exp\left\{\left[J\right]\ T\right\}\ F^{-1}(0) \qquad (43)$$

In this solution, $\Phi(T,0)$ is refered to as the monodromy matrix. The eigenvectors of this monodromy matrix represents the column vectors of the periodic matrix F(0) or F(T). Since $\Phi(T,0)$ and $\exp\left\{\left[J\right]\ T\right\}$ are similar matrices, the eigenvalues of the monodromy matrix $\Phi(T,0)$ are related to the Poincare' exponents through the relationship;

$$\lambda_i = \exp(\omega_i T) \qquad (44)$$

As previously mentioned, Floquet theory just gives information about the form and properties of the solution, and not the solution itself. In order to obtain the solution, the

differential equation for the state transition matrix, Equation (41), must be directly integrated using numerical methods. This integration involves simulating the state transition matrix for one period starting with the appropriate initial conditions of the identity matrix. The topic of numerical methods used for this study is presented in Appendix A.

From a historical point of view, it is interesting to mention the development of helicopter blade stability analyses. Prior to the introduction of the digital computer, the helicopter blade flapping motion was modelled using Hill's equation (21). The solutions to this class of differential equations were precomputed and tabulated assuming a standard form. The blade flapping motion equations were converted to fit this general form, and stability was determined through the use of Floquet theory. This approach generally did not allow for many parametric studies to be conducted and in general, the models themselves had to be less complicated allowing for mathematical simplicity. When the general equations including the aeroelastic effects on the blade were developed (22), the digital computer was used to simulate the motion of the blade in forward motion (23). Stability was determined by simulating the motion of the blade for several revolutions until the growth of the oscillations of the blade could be determined. Later, this technique was replaced by Floquet theory.

The eigenvectors of the monodromy matrix are calculated as follows. The solution is obtained by substituting Equation (43) into Equation (41), which yields;

$$\frac{d}{dt}\left\{F(t)e^{Jt}F^{-1}(0)\right\} = A(t)F(t)e^{Jt}F^{-1}(0) \qquad (45)$$

Carrying through the matrix differentiation of Equation (45) results in the differential equation;

$$\frac{d}{dt} F(t) = A(t)F(t) - F(t)\left[\ J\ \right] \qquad (46)$$

The initial conditions used for the integration of Equation (46) are the eigenvectors of the monodromy matrix. After the integration is completed for one period, the individual values of the elements of matrix F(t) as a function of time are converted into an appropriate Fourier series.

The stability of the dynamical system can be determined through Floquet theory by either looking at the magnitude of the characteristic multipliers (eigenvalues of the monodromy matrix), or the algebraic sign of the associated Poincare' exponents. The conditions for dynamic stability of the system are thus;

$$\sqrt{(\lambda_i(\text{real}))^2 + (\lambda_i(\text{imaginary}))^2} < 1 \qquad (47)$$

or;

$$\omega_i < 0 \qquad (48)$$

Feedback Control Synthesis

For a feedback controller, the system model is described as;

$$\frac{d}{dt} X(t) = A(t)X(t) + B(t)u(t) \tag{49}$$

Making the transformation into Floquet modal form (44);

$$X(t) = F(t)\eta \tag{50}$$

$$\frac{d}{dt} X(t) = \left\{ \frac{d}{dt} F(t) \right\} \eta + F(t) \left\{ \frac{d}{dt} \eta \right\} \tag{51}$$

and substituting Equation (46) into Equation (51) yields;

$$\frac{d}{dt} X(t) = \left\{ A(t)F(t) - F(t)J \right\} \eta + F(t)\frac{d}{dt} \eta \tag{52}$$

Substituting Equations (50) and (52) into Equation (49), the system model in modal form becomes;

$$\frac{d}{dt}\eta = J\eta + F^{-1}(t)B(t)u(t) \tag{53}$$

For a two dimensional case in phase variable state space form, as in Equation (38), the control matrix, B(t), is represented by;

$$B^T = (0 \quad 1) \tag{54}$$

Had the system model been constructed in control canonical form, the control matrix B(t) could have been defined as;

$$(1 \quad 0)^T \text{ or } (1 \quad 1)^T$$

However, in phase variable form, the only definition allowed was that of Equation (54), which represents a torque.

The state feedback term, u(t), is represented by;

$$u(t) = (k_1(t) \quad k_2(t))\eta \tag{55a}$$

$$= K(t)\eta \tag{55b}$$

and;

$$G(t) = F^{-1}(t)B \tag{56a}$$

$$= (g_1(t) \quad g_2(t))^T \tag{56b}$$

The matrix G(t) of equation (56a) represents the internally generated mode controllability matrix of the system model as a result of the product of the inverse eigenvector matrix, $F^{-1}(t)$, and the control matrix B.

In Equation (55b), K(t) represents the gain variables which can be selected by the control designer. Thus, through K(t), the control designer has the ability of single or double modal control.

Single Mode. Assuming that the second modal variable is of interest to control, K(t) takes on the form;

$$\begin{bmatrix} 0 & k_2(t) \end{bmatrix} \tag{57}$$

Substituting this matrix into Equation (53), along with the G(t) matrix as described in Equation (56b), yields the system model in modal form as;

$$\frac{d}{dt}\eta = \begin{bmatrix} \omega_1 & k_2(t)g_1(t) \\ 0 & \omega_2 + k_2(t)g_2(t) \end{bmatrix} \eta \tag{58}$$

In the two dimensional case for scalar control, the selection of the K(t) matrix will either make the matrix of Equation (58) in either upper triangular form, such as it currently is, or in lower triangular form. Approaching Equation (58) as a Floquet type problem, the new Poincare' exponents of the system will be;

$$\omega_1' = \omega_1 \tag{59}$$
$$\omega_2' = \omega_2 + \left(\frac{1}{T}\right)\int_0^T k_2(t)g_2(t)dt \tag{60}$$

Thus, for this particular case, the first Poincare' exponent remains unchanged while the second Poincare' exponent is shifted by the integral;

$$\frac{1}{T}\int_0^T k_2(t)g_2(t)dt \qquad (61)$$

The upper or lower triangular form of Equation (58) will not prevail for higher dimensional systems; however, the sparcity property of the resulting matrix of these systems for single mode control will make the properties as described by Equations (59) and (60) applicable. An example of this can be seen in a third order system where the K(t) matrix is selected as;

$$K(t) = \begin{bmatrix} 0 & k_2(t) & 0 \end{bmatrix} \qquad (62)$$

The modal system differential equation would be;

$$\frac{d}{dt}\eta = \begin{bmatrix} \omega_1 & g_1(t)k_2(t) & 0 \\ 0 & \omega_2 + g_2(t)k_2(t) & 0 \\ 0 & g_3(t)k_2(t) & \omega_3 \end{bmatrix} \eta \qquad (63)$$

The new Poincare' exponent $\omega_2{}'$ is given by;

$$\omega_2{}' = \omega_2 + \left(\frac{1}{T}\right)\int_0^T g_2(t)k_2(t)dt \qquad (64)$$

and the remaining exponents are unchanged.

Again, there is only a change in one Poincare' exponent while the other exponents remain unchanged. However, due to the cross coupling terms, there would be a change in the system

eigenvectors.

The solution of Equation (58) is expressed as;

$$\eta(t) = \eta(0)\exp\int_0^t \begin{bmatrix} \omega_1 & k_2(t)g_1(t) \\ 0 & \omega_2 + k_2(t)g_2(t) \end{bmatrix} dt \qquad (65)$$

<u>Single Scalar Modal Control</u>. The first case to be examined is where the fourier series transformation of the column matrix components resulting from Equation (56a) can be separated into a constant D.C. component and a *periodic* A.C. component. For the present example, this results into the condition;

$$g_1(t) = g_{1\ D.C.} + g_1(t)_{A.C.} \qquad (66)$$

$$g_2(t) = g_{2\ D.C.} + g_2(t)_{A.C.} \qquad (67)$$

Using this expression for $g_1(t)$ and $g_2(t)$, along with a constant gain of k_2 in Equation (57), yields the expression;

$$\eta(t) = \eta(0)\ \exp \begin{bmatrix} \omega_1 & k_2 g_{1\ D.C.} \\ 0 & \omega_2 + k_2 g_{2\ D.C.} \end{bmatrix} t$$

$$x\ \exp\int_0^t \begin{bmatrix} 0 & k_2 g_1(t)_{A.C.} \\ 0 & k_2 g_2(t)_{A.C.} \end{bmatrix} dt \qquad (68)$$

The stability of $\eta(t)$ is determined by the eigenvalues of the matrix;

$$\begin{bmatrix} \omega_1 & k_2 g_1 \text{ D.C.} \\ 0 & \omega_2 + k_2 g_2 \text{ D.C.} \end{bmatrix} \tag{69}$$

These by inspection are;

$$\omega_1 \text{ and } \omega_2 + k_2 g_2 \text{ D.C.}$$

Thus, for scalar control, the new Poincare' exponent location can be achieved through the expression;

$$\omega_2' = \omega_2 + k_2 g_2 \text{ D.C.} \tag{70}$$

For the case where $g_2(t)$ does not have a constant part in its Fourier series, this method is still applicable. For this case, $k_2(t)$ is not considered a constant value, but rather can take on the form, such as (4);

$$k_2(t) = k \sin \frac{2 \eta \pi t}{T} \tag{71}$$

Through the multiplication of $k_2(t) g_2(t)$, the product will generate a constant term through standard trigonometric identities. This term would be kc_{n_2}, where c_{n_2} represents the

n^{th} sine term of the Fourier series expansion of $g_2(t)$. Thus, the new Poincare' exponent would be given as;

$$\omega_2' = \omega_2 + \frac{kc_{n_2}}{2} \tag{72}$$

This procedure would also work if a cosine function was used for Equation (72), where, in this situation, c_{n_2} would represent the n^{th} cosine term in the Fourier series expansion of $g_2(t)$.

Scalar Control of Two or More Modes. This case examines where it is desired to control more than one mode. The approach is applicable for both real or complex Poincare' exponents.

The gain matrix K(t) of Equation (55a) is represented by a row vector composed of individually selected gains per mode;

$$K(t) = \left[k_1(t),\ k_2(t),\ \ldots,\ k_n(t) \right] \tag{73}$$

Substituting K(t) along with Equation (54) and (56b), into Equation (53) yields the result;

$$\frac{d}{dt}\eta = \begin{bmatrix} \omega_1 + g_1(t)k_1(t) & g_1(t)k_2(t) \\ g_2(t)k_1(t) & \omega_2 + g_2(t)k_2(t) \end{bmatrix} \eta \tag{74}$$

This equation is applicable if $k_1(t)$ and $k_2(t)$ represent constant gains, or gains which can be expressed as a sine or

cosine function of the mode and the period. With the gains $g_1(t)$ and $g_2(t)$ represented as a Fourier series, Equation (74) can be written as;

$$\frac{d}{dt}\eta = A1\,\eta + A2(t)\eta \tag{75}$$

where A1 represents a constant matrix and A2(t) represents a matrix with periodic terms.

The shortcoming of this approach is that, by choosing A1 for negative eigenvalues, this selection does not guarantee a stable system (4,42). There is no general way of determining gains such that Equation (75) has predictable Poincare' exponents. With this limitation, the only alternative to control design is to go through the previously described design synthesis of numerically solving the Floquet system to determine the proper gains. However, there is some insights for choice of gains to be offered in this approach.

Using the expression of a matrix derivative from linear algebra for a first order ordinary differential equation;

$$\frac{d}{dt}D(t) = \text{trace}\left\{A(t)\right\}\ D(t) \tag{76}$$

where D(t) represents the determinant of the monodromy matrix, and A(t) represents the matrix of Equation (74). By evaluating the expression for time equalling one period and using Poincare' exponents, the solution to Equation (76) can be

expressed as;

$$\sum \omega_i = \frac{1}{T} \int_0^T \text{trace } A(t)\, dt \qquad (77)$$

For the two dimensional modal case, this expression becomes;

$$\omega_1' + \omega_2' = \omega_1 + \omega_2 + \frac{1}{T} \int_0^T \left\{ k_1(t) g_1(t) + k_2(t) g_2(t) \right\} dt \qquad (78)$$

Assuming that $g_1(t)$ and $g_2(t)$ have constant terms in their Fourier series, constant gains can be used and Equation (78) can be written as;

$$\omega_1' + \omega_2' = \omega_1 + \omega_2 + k_1 g_1 + k_2 g_2 \qquad (79)$$

This expression gives the sum of the new Poincare' exponents but it cannot predict what the individual values of the exponents would be. A systematic design approach thus is required to determine the individual gains k_1 and k_2 to give the desired system performance. It should be mentioned that this approach can be used for the case of complex conjugate exponents to predict the real part of the conjugates.

Recapitulation

For the solution to the control of a time periodic dynamic system, the following steps are taken.

First, the original Floquet system must be solved. The first step to the solution is to solve the monodromy matrix by integrating the dynamic system of equations over one period. From the monodromy matrix, the Poincare' exponents and the stability of the system can be determined. The solution of the monodromy matrix also yields the initial values for the time dependent eigenvectors of the periodic system.

Next, the time dependent eigenvectors are determined from direct integration of Equation (46). This step begins with using the eigenvectors of the monodromy matrix as the initial condition to Equation (46). Equation (46) is then simulated over one period with the eigenvectors evaluated over equally spaced time intervals. Finally, these values are converted into a Fourier series representation of the eigenvectors as a function of time, or as in this study's case, a function of azimuth angle.

When going through a control synthesis of the time periodic dynamic system, a control philosophy must first be selected. The choices of control as presented in this study are scalar control.

The approach to the control design process is to first form the appropriate gain expressions. This is done by forming the mode controllability matrix through Equation (56a), using the information from the eigenvectors and the desired composition of the B matrix. Next, the proper k gain values are selected using the information from the monodromy matrix and

the desired pole locations. Finally, the resulting feedback control law is placed into the simulation model to verify the achievement of the desired pole placement.

Appendix A addresses the variety of numerical methods and computer software used for the individual steps in this design synthesis.

IV. Results

This section will present the study results of the helicopter blade flapping problem using as the mathematical model, Equation (38).

The case for no feedback control will first be explored. Next, a feedback controller for state feedback will be examined. In this case, $\gamma=8$ which is the lock number for a full size blade (18). P^2 was chosen as 1.0 which represents an unconstrained blade. This turns out to be the worst case condition for blade stability. Other values used for this analysis were $\mu = 2.4$, and B=0.97.

No Feedback Case

The models were initially run without feedback for comparisons to other results. For these cases, γ was varied from 2 to 16 by increments of 2. P^2 was varied from 1.0 to 1.8 by step sizes of 0.2 for each γ. In all, this represented 40 cases. The Poincare' exponents were derived for each case using the computer models as described in Appendix A. The results of these runs are presented in Table II. The shaded region of Table IJ represents the cases where the system is unstable. Examples of a stable and unstable response to a step input of one degree is presented in Figure 5.

A comparison of the stability boundary predicted by these cases and those of previous studies (41,36), can be found in Figure 6. As can be seen in this figure, there is close comparisons between the results of this study to other studies.

γ \ P^2		1.0	1.2	1.4	1.6	1.8
2	ω_1	-.4254	-.4302	-.3627	-1.532 + .945i	-1.532 + 1.701i
	ω_2	-.0623	-.0576	-.1250	-1.532 - .945i	-1.532 - 1.701i
4	ω_1	-1.009	-1.002	-.9760	-.9320	-.8680
	ω_2	.0333	.0267	.00035	-.0436	-.1076
6	ω_1	-1.5498	-1.533	-1.507	-1.474	-1.432
	ω_2	.0863	.0698	.0443	.0105	-.0314
8	ω_1	-2.046	-2.026	-2.002	-1.975	-1.936
	ω_2	.0953	.0757	.0506	.0203	-.0148
10	ω_1	-2.512	-2.488	-2.472	-2.482	-2.416
	ω_2	.0819	.0617	.0378	.0102	-.0208
12	ω_1	-2.986	-2.933	-2.954	-2.868	-2.874
	ω_2	.0568	.0375	.0152	-.0098	-.0376
14	ω_1	-3.031	-3.294	-3.030	-3.194	-3.179
	ω_2	.0253	.0075	-.0127	-.0353	-.0602
16	ω_1	-3.336	-3.549	-3.599	-3.639	-3.353
	ω_2	-.0105	-.0263	-.0444	-.0645	-.0865

TABLE II
40 CASES

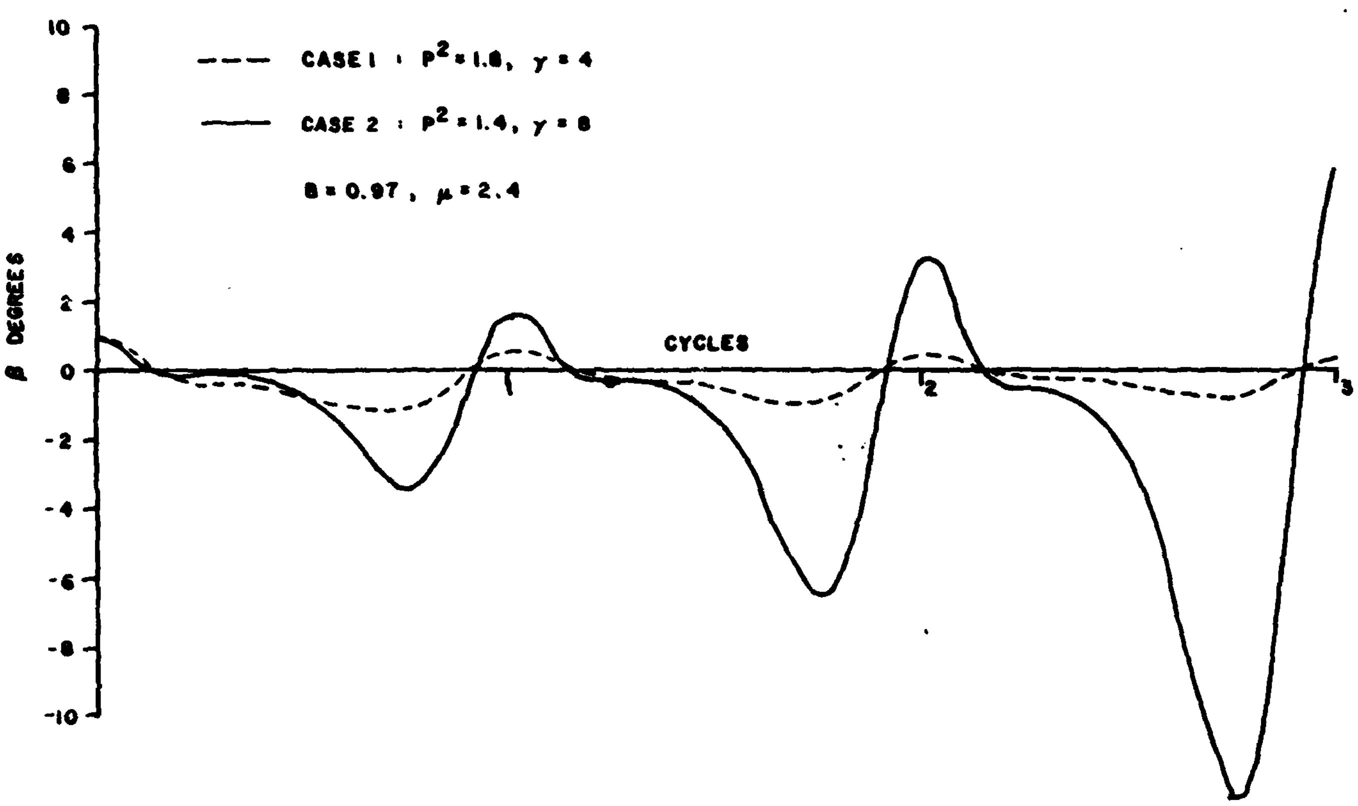


Figure 5. Stable and Unstable Response

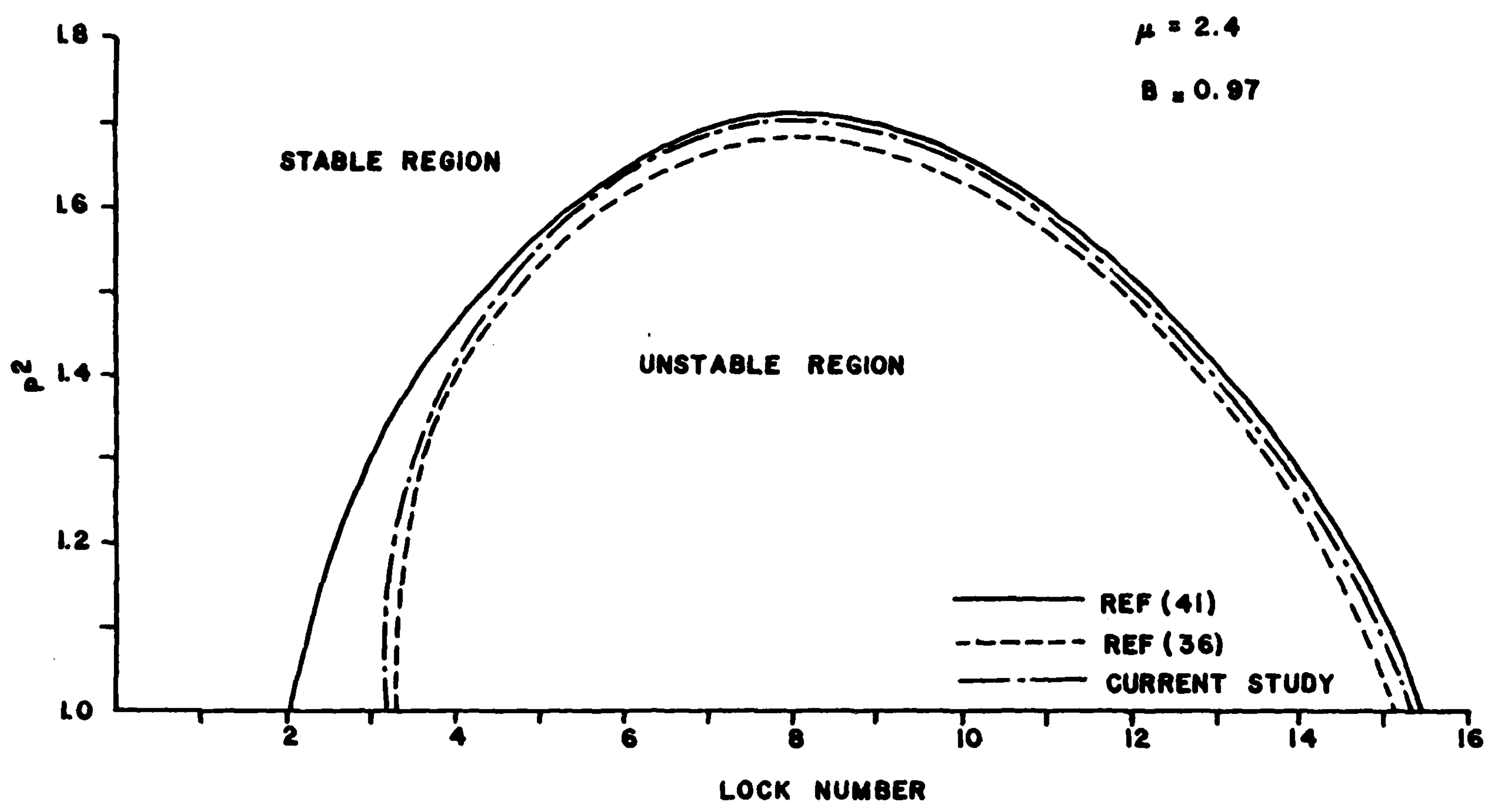


Figure 6. STABILITY BOUNDARY

Feedback Case

As previously mentioned, the condition of $\gamma = 8$ operating with no blade flapping restraint was considered for the feedback case.

For this condition, a root locus plot of the Poincare' exponents with increasing advance ratio is presented in Figure 7. From this plot, it is easily seen that for low advance ratios which correspond to low forward speeds, the roots are complex conjugate pairs. This compares well with the conventional control design approaches where the flapping motion dynamics are modelled as second order systems with complex roots. As the advance ratio is increased, the roots migrate to the real axis, then split with one going to the left (more stable) and the other travelling to the right (less stable). This is why, for the conventional approach to rotor control, gain scheduling is done for the low speed case, then readjusted for higher speeds.

The time dependent eigenvectors for this case at increments of ten degrees are presented in Appendix B.

The Fourier series coefficients of the eigenvector for this case and cases for $\gamma = 2,4,6,8,10,12,14$, and 16, along with their Poincare' exponents and the eigenvalues of the monodromy matrix can be found in Appendix C.

The Fourier coefficients for the elements of the mode controllability matrix for these cases derived by the computer software of Appendix I can be found in Appendix D.

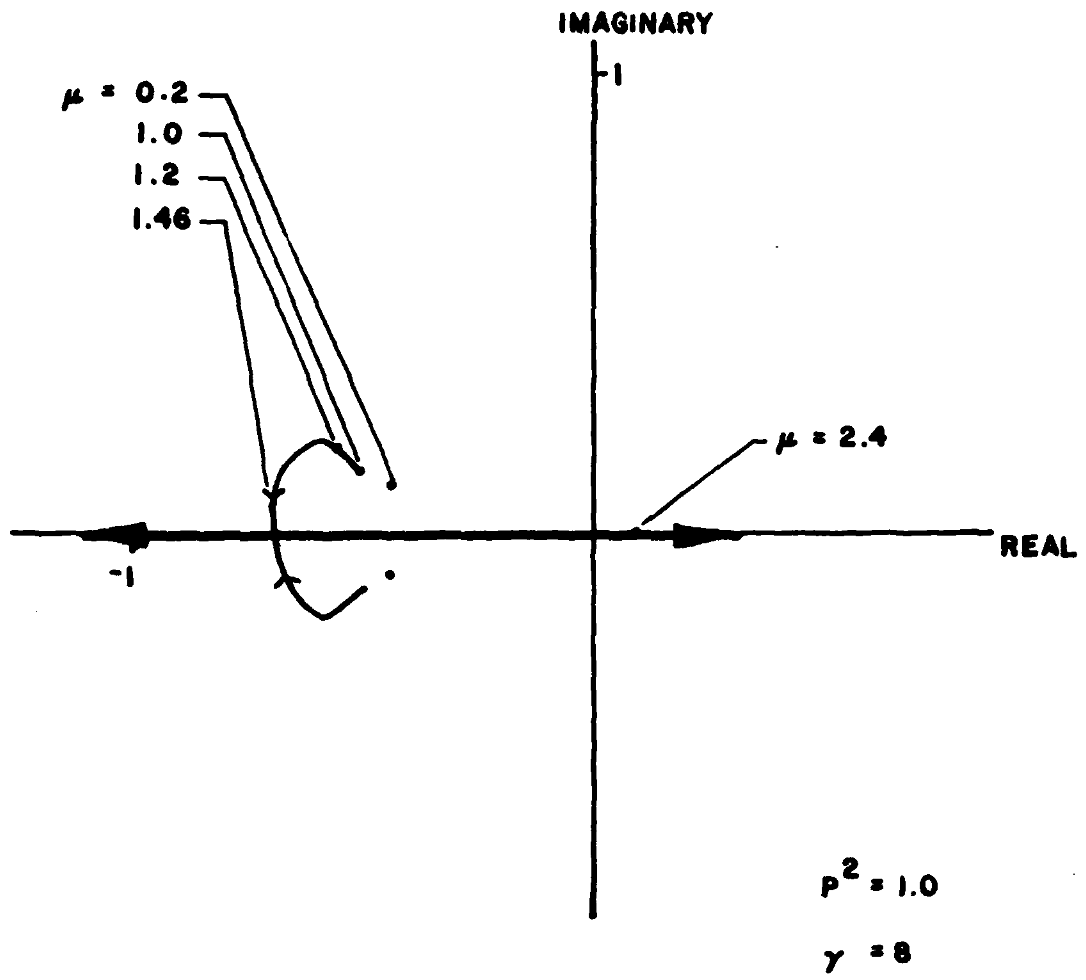


Figure 7. Root Locus

For $\gamma = 8$, the second Poincare' exponent was changed from its present unstable positive value to zero. Using the constant A_o terms from Appendix D and Equation (70), the required gain to accomplish this task was calculated as;

$$B^T = (\ 0 \quad 1\ ),\ k = 0.109737$$

Placing these values into a feedback control simulation (Appendix J), the resulting Poincare' exponents were;

$$\omega_1 = -2.050636$$
$$\omega_2 = 0.000239$$

As can be seen by these results, desired pole placement of ω_2 was obtained with minor change to ω_1. The reason for the slight change is due to the Runga-Kutta integration routine with associated roundoff error and step size.

A comparison of the response of this system to the case of no feedback can be found in Figure 8.

By applying this gain to the other cases with different lock numbers, the system still remains stable. This was done by using the A_o terms from Appendix D and the Poincare' exponents from Table II. The comparison of the old Poincare' exponents to the achieved exponents is found in Table III.

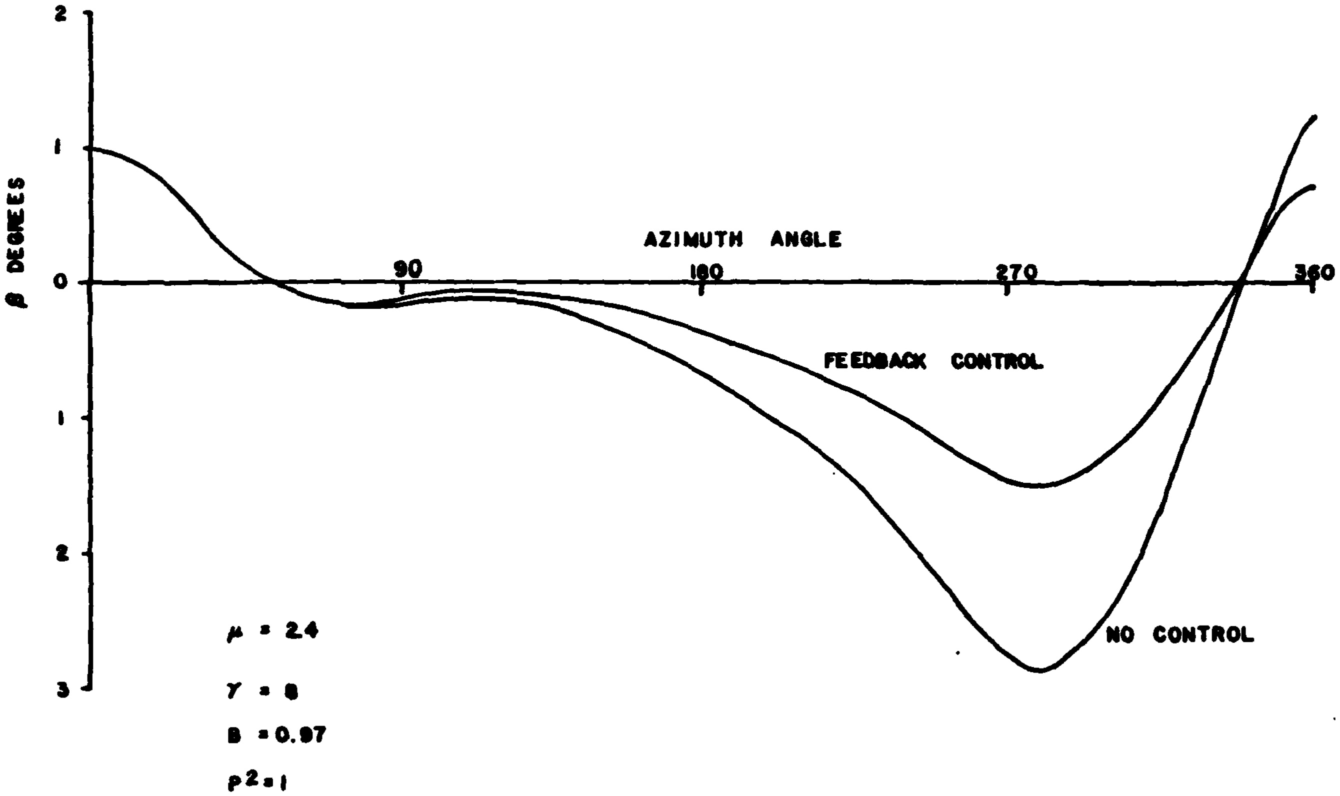


Figure 8. Feeback Control Performance

γ	2	4	6	8	10	12	14	16
ω_2	-.062333	.033374	.086369	.095386	.081904	.056859	.02530	-.010564
ϕ_2	-.335545	-.68878	-.79112	-.86922	-.938705	-1.46332	-9.5705	-68.602
ω'_2	-0.099155	-0.042211	-.000446	0	-.021107	-.103721	-1.024938	-7.538742

TABLE III
POINCARE' EXPONENT SHIFT

V. Conclusion

From this study, it was shown how the principles of Floquet theory could be applied to the helicopter blade flapping control design synthesis.

By first deriving the appropriate blade flapping equations of motion for a rigid blade with no twist, a series of computer simulations were assembled. Each of these simulations addressed a different aspect in solving the Floquet problem.

From the analysis of a blade with Lock number of 8, it was shown how conveniently a feedback controller could be designed for the case of state feedback. The conventional approach to the same problem would begin the design process by first considering low forward flight, then define stability regions for different forward velocities. The gains would then have to be readjusted until the system was in a stable region. This step has been eliminated with Floquet theory application.

This has been the first step in the process of realizing the full control scheme for the helicopter rotor. Further topics which need to be addressed are the coupled blade flapping with lead-lag degrees of freedom. This is another well suited application of Floquet theory control design synthesis.

Another case which renders itself to this type of analysis is the case where the blade is travelling through atmospheric turbulence. In this condition, the flow model as described in Section II would be replaced with a stochastic model.

Appendix A. Numerical Methods

As presented in Section II and Section III, numerical methods are required to derive the solution of the time periodic blade flapping equation of motion. These methods are required due to the fact that Floquet theory can only give information about the form and properties of the solution, but not the solution itself. To obtain the solution, the dynamic system must be simulated over one period to obtain the proper data to derive the monodromy matrix, the periodic eigenvectors of the system as a Fourier series representation dependent on azimuth angle, and the periodic gain properties as a Fourier series representation dependent on azimuth angle.

The required algorithms to do this analysis, as highlighted in Section III, consist of an integration routine, eigenproblem routine, and a Fourier series coefficient determination routine.

The purpose of this section is to present the software developed to handle this Floquet type problem. The software developed can be found in Appendix E,F,G,H,J,and J, and will be described by highlighting the specific function of each program. All the programs have been written in Texas Instruments Extended BASIC, which is a dialect of the BASIC programming language, and is closely related to Microsoft BASIC-80.

The computer used to develop and conduct the research of this study was the Texas Instrument 99-4A microcomputer.

This computer uses the TMS9900 16-bit microprocessor which utilizes a 16-bit data bus. The microprocessor as configured can give a decimal precision of 13 to 14 digits depending on the numerical value, and an exponential decimal range from -128 to +126. These capabilities make this computer suitable to handle the numerical accuracies as demanded in the eigenproblem algorithm. Other required computer hardware to conduct this research consisted of 48K-bytes of random access memory, and a disk drive for program storage and data manipulation.

Integration Routines

Two integration routines were required to derive the solution to Equation (41) and Equation (46). Both routines use the blade flapping motion model as described in Equation (38).

The integration technique employed in both routines was a fourth-order Runga-Kutta routine (37:194,40:182). The Runga-Kutta routine was selected because of its numerical accuracy and self-starting capability. Other higher order Runga-Kutta techniques could have been chosen; however, a tradeoff was made between accuracy and computer running time. Both routines integrated with respect to azimuth angle as dictated by Equation (38), with the step size of integration adjusted to give the best accuracy. The adjustment of integration step size will be presented later in this section.

The first integration routine (Appendix E), was used to

determine the monodromy matrix described by Equation (41). The function of the program and its subroutines follow.

The executive portion of the program first initializes all variables used. Afterwards, it calls for the parameters of tip loss factor, advance ratio, Lock number, and the natural frequency of the helicopter blade in hover. From there, it begins the fourth-order Runga-Kutta routine to simulate the flapping motion of the blade for one revolution. Afterwards, the numerical values of the elements of the monodromy matrix are printed out.

Subroutine FCNN was the routine which represented Equation (41). Major calls from this subroutine are to subroutine MATRIXA, which represents the helicopter blade model of Equation (38), and subroutine MATRIXMULT, which is a standard matrix multiplication routine. Subroutine FCNN was called from the imbedded Runga-Kutta routine of the executive program.

Subroutine MATRIXA contains the helicopter blade model. Imbedded in this subroutine where the equations for the periodic coefficients C(ψ) and K(ψ) (lines 2022-2035), along with the determination of the flow region (lines 2010-2016). The actual matrix evaluation of Equation (38) was conducted in lines 2052-2055. The subroutine returns updated values to subroutine FCNN.

Subroutine MATRIXMULT was a standard matrix multiplication routine (31:41). The subroutine was called from subroutine FCNN, and returns its values to the same subroutine.

The next integration routine used in this analysis can

be found in Appendix G. The purpose of this routine was to solve Equation (46). This program also uses an imbedded fourth-order Runga-Kutta routine.

The executive portion of this routine was structured similarily to that of the last program. The program first initializes all variables. The inputs used in the program were the same as the last program with the addition of the Poincare' exponents and the eigenvectors of the monodromy matrix. The output of this routine were the elements of the eigenvector matrix for every ten degrees of azimuth angle. This output is later used in the Fourier series coefficient analyses.

Subroutine FCNN represents the right hand side of Equation (46). This subroutine is called per each step of the Runga-Kutta from the executive program. The main calls from this subroutine were to subroutine MATRIXA, which was previously described, subroutine MATRIXMULT which was also described, and subroutine MATRIXADD, which is a standard matrix addition routine.

Since both subroutines MATRIXA and MATRIXMULT have been described, the details of these subroutines will not be presented here.

Subroutine MATRIXADD is a standard matrix addition routine (31:40). No further details on this subroutine will be presented.

Subroutine INPUTT is the routine which handles the matrix input as required by the executive program.

Fourier Series Coefficients Analysis Routines

The two Fourier series analysis routines used in this research can be found in Appendix H and I. Both routines use the eigenvector matrix data for every ten degrees of azimuth angle as computed by the previously described program. The output of both programs are the first 19 Fourier coefficients for each of their respected cases.

The definition of the Fourier series used in both programs are mathematically described in the following equation (30:479, 19:34,40:148);

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{18} \left[a_k \cos \frac{2\pi tk}{T} + b_k \sin \frac{2\pi tk}{T} \right] \qquad (80)$$

The Fourier series coefficients are described by the following equations;

$$a_k = \frac{2}{T}\int_0^T f(t)\cos \frac{2\pi tk}{T}\, dt \quad k=0,1,2,3,\ldots,18 \qquad (81)$$

$$b_k = \frac{2}{T}\int_0^T f(t)\sin \frac{2\pi tk}{T}\, dt \quad k=1,2,3,\ldots,18 \qquad (82)$$

As can be seen in Equations (81) and (82), an integration routine is also required in the software. Because of its accuracy, a Simpson integration routine (19:25) was imbedded in both programs. It is important to note that the discrete form of the Fourier series analysis was not used due to its problem of distributing the Fourier coefficients along all the harmonics of the Fourier series. The continuous model

does not have this problem which now, allows for the elimination of those harmonics with lower order coefficients. Both methods, however, will give the same value of the constant component of the Fourier series which is of prime interest in the Floquet analysis.

The purpose of the first Fourier analysis program (Appendix H) is to derive the Fourier series representation of the eigenvector matrix for the homogeneous Floquet problem as described by Equation (42). The program consist of two parts, the executive routine which handles the data input and output functions along with the final stages of the Fourier analysis routine, and subroutine FOURIER which handles the summation and integration of the data.

As previously mentioned, the executive routine handles both data manipulation tasks and completes the Fourier series analysis. Lines 570 and 580 computes the nineteen a and b Fourier coefficients by multiplying the summation terms from subroutine FOURIER by the Simpson integration constant;

$$\frac{\Delta\psi}{3} \qquad (83)$$

and the Fourier series constant;

$$\frac{2}{T} \qquad (84)$$

The increment of azimuth angle used in this analysis was ten degrees with the period of the system set at 360 degrees.

Subroutine FOURIER performs the integration of the data as dictated by Equations (81) and (82). Lines 960 and 980 are the locations in the routine where this function takes place. These two lines are also the location where the Simpson integration takes place through the multiplication of each Fourier product by the term CONSTANT. The term CONSTANT represents the coefficients 1 for the first and last data entry, 4 for odd numbered data entries, and 2 for even number data entries. The determination of these terms is conducted in lines 912-916.

The purpose of the second Fourier series analysis routine (Appendix I), is to evaluate the mode controllability matrix as mathematically described by Equation (56a). Although the program is written with three options for the B matrix, the B matrix used in this analysis was;

$$B^T = (0 \quad 1) \tag{85}$$

The structure of this program is very similar to that of the previously described program with the addition of a matrix multiplication routine and a matrix inversion routine. The input to this program is also identical to the last program. The Fourier analysis routines are identical.

The difference in the executive portion of the program is the multiplication of the inverted eigenvector matrix with the B matrix. The product of this multiplication is then fed into the Fourier analysis routine. This is accomplished in

the program at line locations 442-449 and 492-499. The added subroutines in this program are subroutine MATRIXMULT, which has been described before, and the matrix inversion routine INVERSE.

Subroutine INVERSE conducts the matrix inversion of the eigenvector matrix through a Gauss-Jordan routine (32;78). The inversion is conducted by rotating the input matrix about its diagonal, then normalizing the result. It is interesting to note that for the k gains not to achieve an infinite value, the controllability criteria for the system must be satisfied for all time increments. Subroutine INVERSE will stop the execution of the program if this criteria is not met.

<u>Eigenproblem Routine</u>

The eigenproblem routine (Appendix F) was a converted routine originally written in FORTRAN 4 (17). This program provides real or complex eigenvalues and eigenvectors for a real matrix in general form. The program first reduces the matrix to an upper Hessenberg form by means of a Householder transformation technique. Then, a unitary triangular transformation (known as a QR transformation) is conducted on the converted Hessenberg matrix until all elements of the subdiagonal converge to a value which can be approximated to equal zero. The eigenvalues are then determined from this reduced form. Inverse iteration is performed on the upper Hessenberg matrix until the absolute value of the largest component of the right hand side vector is greater than the bound of 100 (a bound established by the numerical accuracy performance of the

computer used). This computation yields the eigenvectors of the matrix. A brief description of the function of each routine follows.

The executive routine of the program performs the data management task of the program. It first accepts the input of the monodromy matrix as provided from the program of Appendix E. Next, it normalizes the matrix through a call to subroutine SCALEE. Following this, it begins to determine the eigenvalues of the normalized matrix by a call to subroutine HESQR. From the eigenvalues, the executive program determines if the eigenvectors will be either real or complex, and calls the appropriate subroutine. Finally, it reconstructs the original matrix and converts the eigenvectors to this matrix. After the eigenvectors are normalized, the eigenvalues and eigenvectors are printed out.

Subroutine SCALEE performs the scaling function of the original matrix. The array is scaled so that the quotient of the absolute sum of the off diagonal terms of each column and the absolute sum of the off diagonal elements of each row lies within the bounds computed in the program. After the matrix is scaled, it is normalized so that the value of the Euclidian norm is equal to one. The scaling factor computed and used in this routine is stored for later use in reconstructing the original matrix.

Subroutine HESQR determines the eigenvalues of the real general matrix. The original matrix is converted to an upper Hessenberg form using the Householder transformation method.

The transformation process to do this conversion uses a QR approach. The end result of this routine is the indication of whether the eigenvalues were determined, the eigenvalues, and a flag to signify whether the eigenvalues are real or complex.

Subroutine REALVE determines the real eigenvectors of the real eigenvalues of the matrix. This is performed using an inverse iteration method which employs a Gaussian elimination technique. Afterwards, backsubstitution is performed until the eigenvector matrix is obtained.

Subroutine COMPVE determines the complex eigenvectors of the complex eigenvalues of the matrix. This routine splits the complex eigenvalue matrix into two matrices containing the real parts and imaginary parts of the eigenvalues. It then uses an inverse iteration technique, also employing a Gaussian elimination procedure, until the complex eigenvectors are determined.

Subroutine INPUTT was added to this routine to handle the input of the monodromy matrix to the program.

<u>Feedback Routine</u>

The last program used in this analysis can be found in Appendix J and represents the system with feedback. The structure of this routine is very similar to the program of Appendix E with the addition of the feedback loop incorporated in subroutine FCNN. This addition is found in lines 1490-1560 of the program listing. The predetermined k gain value and the B matrix is inputted in lines 1450-1470.

Step Size Determination

As the case for all digital simulations of dynamic systems, the step size selected for the integration routines will have a major impact on the results and the inferred stability determination derived from these results. Because of the property of the one sample time delay characteristic in all digital simulations, a step size chosen too large would infer stability when in fact, stability does not exist in the continuous dynamic system.

Another important consideration in any digital simulation, which has a major impact for the helicopter blade model, is selecting a sample size so as to capture all the environmental factors which would impact the stability of the system. For the helicopter problem, the effects of the various flow regions as described in Section II are of primary concern. The width of the mixed flow region for the rotor is 2.8 degrees for the conditions run for this study. A step size too large would completely miss this region and would thus lead to erroneous results. For this region, it is highly desirable to have several "hits" in this region to realize the effects of mixed flow on the flapping motion of the blade.

Keeping these two concerns in mind, the monodromy matrix determination program of Appendix E was run for the various azimuth degree increments of 45,30,20,10,5,2,1, and 0.5 degrees. The resulting matrices from these runs were then inputted into the eigenprogram routine of Appendix F. The resulting eigenvalues were compared to determine convergency

of the roots. The results of this analysis can be found in Table IV.

As seen in Table IV, the eigenvalues converge to a constant value at a degree increment of one degree. Thus, an integration step size of one degree was selected for both the monodromy matrix determination and the eigenvector determination programs. This selection led to a program execution time of 41 minutes.

Because the Simpson integration routine used in the Fourier series analysis programs uses data from a one degree step size Runga-Kutta routine, ten degree increment step sizes were felt appropriate for these routines.

Analysis Procedure

The sequential use of each program used for this research is as follows:

First, the monodromy matrix was determined using the program of Appendix E, with the appropriate input values.

Next, the monodromy matrix was inputted into the eigenproblem program of Appendix F. This gave the eigenvalues and eigenvectors of the monodromy matrix. The eigenvalues were then converted to the Poincare' exponents to be used in the next program.

With the information of the Poincare' exponents, the eigenvectors of the monodromy matrix along with the operating parameters used in the first program were used in the program of Appendix G to determine the eigenvector components for every ten degrees of azimuth angle.

$\Delta\psi°$	λ_1	λ_2
45	3.9537×10^{-3}	7.0426×10^{-1}
30	2.0605×10^{-3}	1.1806
20	1.8075×10^{-3}	1.2286
10	1.7661×10^{-3}	1.2334
5	1.7646×10^{-3}	1.2334
2	1.8075×10^{-3}	1.2286
1	1.7643×10^{-3}	1.2333
.5	1.7643×10^{-3}	1.2333

TABLE IV. Step Size Sensitivity Run

Finally, the eigenvector information was fed into the two Fourierseries analysis programs of Appendix H and I to determine the Fourier series representation of the eigenvector matrix and mode controllability matrix.

Using the information from all the programs run, the desired gain value for k was determined for eigenvalue placement. This value of k along with the appropriate B matrix was inputted into the feedback control simulation of Appendix J to verify this placement.

Appendix B.

F Matrix

INCREMENT OF RUN EQUALS 36 STEPS
FOR THIS RUN;
B= .97 MU= 2.40 GAMMA= 8.00 PSQUARE= 1.00

POINCARE EXPONENTS

OMEGA 1=-20.36572E-01 OMEGA 2= 95.58700E-03

SYSTEM OF EIGENVECTORS

ANGLE= 0

20.57116E-02	43.25090E-02
-97.86127E-02	90.16296E-02

ANGLE= 10

34.90659E-02	52.35988E-02
69.81317E-02	87.26646E-02

ANGLE= 20

12.21730E-01	13.96263E-01
15.70796E-01	17.45329E-01

ANGLE= 30

-77.00381E-03	15.31811E-01
-19.38746E-03	19.19862E-01

ANGLE= 40

22.68928E-01	24.43461E-01
26.17994E-01	27.92527E-01

ANGLE= 50

31.41593E-01	33.16126E-01
34.90659E-01	-14.45449E-02

ANGLE=	60
51.57152E-03	-40.62317E-02
36.65191E-01	38.39724E-01

ANGLE=	70
41.88790E-01	43.63323E-01
45.37856E-01	47.12389E-01

ANGLE=	80
50.61455E-01	52.35988E-01
-61.75976E-01	-92.39686E-02

ANGLE=	90
93.06056E-02	54.10521E-01
55.85054E-01	57.59587E-01

ANGLE=	100
61.08652E-01	62.83185E-01
-15.60598E-01	-19.13540E-03

ANGLE=	110
23.83414E-02	-80.18499E-03
-10.60967E-01	-35.70454E-03

ANGLE=	120
15.70344E-02	-87.63131E-03
-68.99220E-02	-67.51602E-03

ANGLE= 130

10.62086E-02	-10.14263E-02
-44.06939E-02	-11.02469E-02

ANGLE= 140

76.24102E-03	-12.33370E-02
-28.29233E-02	-16.40636E-02

ANGLE= 150

60.02919E-03	-15.52158E-02
-18.59474E-02	-22.95033E-02

ANGLE= 160

53.11715E-03	-19.87481E-02
-12.71800E-02	-30.38605E-02

ANGLE= 170

52.95849E-03	-25.45764E-02
-93.23645E-03	-37.75201E-02

ANGLE= 180

57.86364E-03	-32.08296E-02
-79.34749E-03	-43.08191E-02

ANGLE= 190

65.59487E-03	-39.14462E-02
-88.72547E-03	-43.70415E-02

ANGLE= 200

72.16231E-03	-45.80382E-02
-12.22973E-02	-40.60103E-02

ANGLE= 210

73.76564E-03	-52.05951E-02
-15.71218E-02	-41.87212E-02

ANGLE= 220

70.62701E-03	-59.15475E-02
-17.24072E-02	-51.07524E-02

ANGLE= 230

64.68776E-03	-68.13224E-02
-17.11036E-02	-64.08809E-02

ANGLE= 240

57.46074E-03	-79.13331E-02
-16.13522E-02	-75.15478E-02

ANGLE= 250

49.31963E-03	-91.25286E-02
-15.14399E-02	-78.12904E-02

ANGLE= 260

39.29270E-03	-10.25840E-01
-14.69405E-02	-67.65547E-02

ANGLE= 270

25.22115E-03	-11.05226E-01
-14.91571E-02	-41.01751E-02

ANGLE= 280

42.62420E-04	-11.24141E-01
-15.39795E-02	-10.17435E-04

ANGLE= 290

-26.19652E-03	-10.64337E-01
-15.13664E-02	48.00967E-02

ANGLE= 300

-67.28986E-03	-92.34002E-02
-12.67624E-02	93.00678E-02

ANGLE= 310

-11.74665E-02	-71.67134E-02
-65.39768E-03	12.54430E-01

ANGLE= 320

-17.18765E-02	-47.19343E-02
41.75862E-03	14.08173E-01

ANGLE= 330

-22.26130E-02	-21.82252E-02
19.42004E-02	14.13980E-01

ANGLE= 340

-25.92404E-02	24.96956E-03
38.54997E-02	13.50662E-01

ANGLE= 350

-26.66743E-02	24.92593E-02
63.20230E-02	12.24733E-01

ANGLE= 360

-21.78727E-02	43.27003E-02
95.39180E-02	90.15412E-02

Appendix C.

Fourier Series Coefficients of F Matrices

INCREMENT OF RUN EQUALS 360 STEPS

FOR THIS RUN;
B= .97 MU= 2.40 GAMMA= 2.00 PSQUARE= 1.00

POINCARE EXPONENTS

OMEGA1=-42.54970E-02 OMEGA2=-62.33300E-03

SYSTEM OF EIGENVECTORS

16.5818E-02	31.1379E-02
-98.6156E-02	95.0286E-02

A 0 COEFFICIENTS

-19.6762E-02	-44.8445E-02
83.7176E-03	27.9506E-03

B 0 COEFFICIENTS

.0000E+00	.0000E+00
.0000E+00	.0000E+00

A 1 COEFFICIENTS

16.1161E-02	43.8081E-02
-59.7067E-02	52.9658E-02

B 1 COEFFICIENTS

-52.8504E-02	55.6940E-02
63.7176E-03	-47.2800E-02

A 2 COEFFICIENTS

58.6173E-03	10.4478E-02
-42.6056E-02	26.9492E-02

B 2 COEFFICIENTS

-20.0555E-02	13.8007E-02
-31.8989E-03	-21.7567E-02

A 3 COEFFICIENTS

37.0986E-03	97.6622E-04
-52.4787E-03	13.1302E-02

B 3 COEFFICIENTS

-12.2395E-03	43.9555E-03
-10.6081E-02	-32.0435E-03

A 4 COEFFICIENTS

54.9508E-04	-12.3290E-03
23.7879E-03	-61.9793E-04

B 4 COEFFICIENTS

65.3177E-04	-17.3308E-04
-24.7809E-03	49.3626E-03

A 5 COEFFICIENTS

13.8490E-04	-35.0863E-04
19.5926E-03	11.1942E-03

B 5 COEFFICIENTS

40.2743E-04	21.8504E-04
-86.2134E-04	17.4278E-03

A 6 COEFFICIENTS

14.8762E-05	-66.2052E-05
23.0512E-04	-39.5225E-04

B 6 COEFFICIENTS

39.1465E-05	-65.3117E-05
-11.6032E-04	38.0558E-04

A 7 COEFFICIENTS

88.5913E-06	-35.1997E-05
23.4672E-04	36.7084E-04

B 7 COEFFICIENTS

33.7539E-05	53.2866E-05
-74.4634E-05	25.6373E-04

A 8 COEFFICIENTS

10.9356E-05	83.3360E-06
-26.9385E-05	-33.3286E-05

B 8 COEFFICIENTS

-38.7572E-06	-28.7435E-06
-11.7023E-04	-11.4574E-04

A 9 COEFFICIENTS

72.1498E-07	-59.5302E-06
24.4448E-05	74.3820E-05

B 9 COEFFICIENTS

55.4548E-06	16.5221E-05
-15.6874E-05	10.5438E-04

A10 COEFFICIENTS

10.9672E-06	46.1894E-06
-58.2987E-06	25.7881E-05

B10 COEFFICIENTS

-25.6176E-06	54.5014E-07
-84.9673E-05	-11.0347E-04

A11 COEFFICIENTS

-32.6572E-06	97.0767E-06
-82.2738E-05	-11.0370E-04

B11 COEFFICIENTS

10.8318E-05	18.7380E-05
-21.4283E-05	10.6874E-04

A12 COEFFICIENTS

-34.1523E-06	25.2367E-05
-80.8118E-05	14.5163E-04

B12 COEFFICIENTS

12.8032E-05	-20.5924E-05
-56.1285E-05	90.4627E-05

A13 COEFFICIENTS

-46.4061E-05	11.6634E-04
-66.1746E-04	-38.3395E-04

B13 COEFFICIENTS

13.3460E-04	71.7985E-05
-28.3837E-04	58.4432E-04

A14 COEFFICIENTS

-18.2899E-04	41.1753E-04
-79.2725E-04	21.3387E-04

B14 COEFFICIENTS

21.7785E-04	-57.1831E-05
-82.9825E-04	16.3374E-03

A15 COEFFICIENTS

-12.3667E-03	-32.5420E-04
17.4421E-03	-43.8616E-03

B15 COEFFICIENTS

-40.8463E-04	14.6429E-03
-35.3442E-03	-10.6961E-03

A16 COEFFICIENTS

-19.5407E-03	-34.8281E-03
14.2029E-02	-89.8134E-03

B16 COEFFICIENTS

-66.8509E-03	46.0046E-03
-10.6292E-03	-72.5378E-03

A17 COEFFICIENTS

-53.7198E-03	-14.6025E-02
19.9005E-02	-17.6603E-02

B17 COEFFICIENTS

-17.6170E-02	18.5644E-02
21.2431E-03	-15.7610E-02

A18 COEFFICIENTS

65.5847E-03	14.9477E-02
-27.8947E-03	-93.1753E-04

B18 COEFFICIENTS

-39.1895E-11	16.6863E-11
38.5025E-11	-80.9785E-11

INCREMENT OF RUN EQUALS 360 STEPS

FOR THIS RUN;

B= .97 MU= 2.40 GAMMA= 4.00 PSQUARE= 1.00

POINCARE EXPONENTS

OMEGA1=-10.09043E-01 OMEGA2= 33.37400E-03

SYSTEM OF EIGENVECTORS

19.5071E-02	27.3698E-02
-98.0789E-02	96.1816E-02

A 0 COEFFICIENTS

-30.0984E-02	-52.5723E-02
30.3711E-02	-17.5398E-03

B 0 COEFFICIENTS

.0000E+00	.0000E+00
.0000E+00	.0000E+00

A 1 COEFFICIENTS

92.8545E-03	30.0678E-02
-46.6463E-02	49.0367E-02

B 1 COEFFICIENTS

-37.2792E-02	48.0288E-02
28.3306E-02	-28.4656E-02

A 2 COEFFICIENTS

14.2285E-02	19.9223E-02
-61.3334E-02	26.4648E-02

B 2 COEFFICIENTS

-23.4882E-02	12.9002E-02
-47.5636E-03	-39.4158E-02

A 3 COEFFICIENTS

87.9682E-03	57.0997E-03
-15.7750E-02	14.5492E-02

B 3 COEFFICIENTS

-23.0123E-03	47.8377E-03
-24.0689E-02	-16.9718E-02

A 4 COEFFICIENTS

21.3184E-03	-10.5508E-03
32.3494E-03	24.7797E-03

B 4 COEFFICIENTS

13.4669E-03	62.9339E-04
-98.8990E-03	42.3102E-03

A 5 COEFFICIENTS

16.4395E-04	-69.8895E-04
52.7459E-03	36.0357E-03

B 5 COEFFICIENTS

10.8635E-03	72.3913E-04
-19.1835E-03	35.1926E-03

A 6 COEFFICIENTS

-40.1737E-05	-21.9559E-04
13.3768E-03	36.5349E-05

B 6 COEFFICIENTS

21.6573E-04	93.7120E-06
43.6855E-06	12.8428E-03

A 7 COEFFICIENTS

-37.8287E-05	-85.0409E-05
59.4076E-04	63.5882E-04

B 7 COEFFICIENTS

78.7641E-05	93.5560E-05
18.5991E-04	61.3462E-04

A 8 COEFFICIENTS

12.6770E-05	54.5490E-06
25.7955E-05	-17.7996E-05

B 8 COEFFICIENTS

55.8909E-06	78.0273E-07
-16.2696E-04	-11.7237E-04

A 9 COEFFICIENTS

-10.7635E-06	-90.0110E-06
44.0501E-05	12.4261E-04

B 9 COEFFICIENTS

95.7118E-06	27.7400E-05
93.6313E-06	16.2429E-04

A10 COEFFICIENTS

42.5617E-06	95.3395E-06
-67.3533E-06	43.4979E-05

B10 COEFFICIENTS

28.6337E-06	39.2950E-06
-13.8270E-04	-14.9439E-04

A11 COEFFICIENTS

12.4900E-05	26.0796E-05
-20.5656E-04	-18.7068E-04

B11 COEFFICIENTS

25.4520E-05	33.2638E-05
63.7637E-05	22.8709E-04

A12 COEFFICIENTS

16.1772E-05	78.2937E-05
-44.3213E-04	10.9768E-05

B12 COEFFICIENTS

72.6581E-05	50.5044E-06
-30.4941E-05	36.9516E-04

A13 COEFFICIENTS

-54.7486E-05	23.2860E-04
-17.7424E-03	-12.1545E-03

B13 COEFFICIENTS

36.0658E-04	23.9805E-04
-63.8731E-04	11.7349E-03

A14 COEFFICIENTS

-71.0133E-04	35.3035E-04
-10.7607E-03	-81.6823E-04

B14 COEFFICIENTS

44.9143E-04	21.0518E-04
-33.0330E-03	13.9053E-03

A15 COEFFICIENTS

-29.3214E-03	-19.0306E-03
52.4914E-03	-48.6446E-03

B15 COEFFICIENTS

-76.7956E-04	15.9320E-03
-80.2335E-03	-56.6128E-03

A16 COEFFICIENTS

-47.4314E-03	-66.4106E-03
20.4456E-02	-88.2039E-03

B16 COEFFICIENTS

-78.2929E-03	43.0028E-03
-15.8456E-03	-13.1418E-02

A17 COEFFICIENTS

-30.9499E-03	-10.0223E-02
15.5457E-02	-16.3540E-02

B17 COEFFICIENTS

-12.4267E-02	16.0091E-02
94.4321E-03	-94.9038E-03

A18 COEFFICIENTS

10.0324E-02	17.5234E-02
-10.1231E-02	58.3387E-04

B18 COEFFICIENTS

-45.4389E-11	33.9618E-12
62.4860E-11	-91.4755E-11

INCREMENT OF RUN EQUALS 360 STEPS

FOR THIS RUN;
B= .97 MU= 2.40 GAMMA= 6.00 PSQUARE= 1.00

POINCARE EXPONENTS

OMEGA1=-15.49838E-01 OMEGA2= 86.36900E-03

SYSTEM OF EIGENVECTORS

21.2337E-02	35.1497E-02
-97.7197E-02	93.6189E-02

A 0 COEFFICIENTS

-32.2339E-02	-54.3168E-02
49.9578E-02	-46.8978E-03

B 0 COEFFICIENTS

.0000E+00	.0000E+00
.0000E+00	.6000E+00

A 1 COEFFICIENTS

21.6828E-03	26.6427E-02
-36.2954E-02	49.9360E-02

B 1 COEFFICIENTS

-32.9368E-02	47.6285E-02
48.8778E-02	-22.5302E-02

A 2 COEFFICIENTS

17.6584E-02	26.9833E-02
-75.3277E-02	22.0735E-02

B 2 COEFFICIENTS

-23.9800E-02	98.7114E-03
18.4834E-03	-53.1168E-02

A 3 COEFFICIENTS

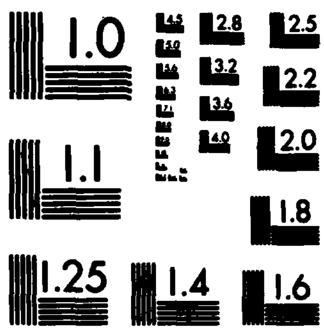
13.3284E-02	98.7115E-03
-28.7070E-02	89.0562E-03

B 3 COEFFICIENTS

-26.8498E-03	26.8068E-03
-35.8259E-02	-29.3855E-02

AD-A154 460 APPLICATION OF FLOQUET THEORY TO HELICOPTER BLADE FLAPPING STABILITY(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING 2/3

UNCLASSIFIED J K MARCH DEC 84 AFIT/GAE/AA/84D-13 F/G 1/3 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

A 4 COEFFICIENTS

42.1554E-03	-11.8375E-04
33.9278E-03	37.9430E-03

B 4 COEFFICIENTS

24.8203E-03	95.1989E-04
-20.7133E-02	53.9532E-04

A 5 COEFFICIENTS

31.9844E-04	-65.2583E-04
93.4039E-03	74.7848E-03

B 5 COEFFICIENTS

19.6558E-03	15.0479E-03
-46.4943E-03	33.8744E-03

A 6 COEFFICIENTS

-19.8555E-04	-29.7538E-04
35.5153E-03	17.9749E-03

B 6 COEFFICIENTS

54.2028E-04	30.6696E-04
33.0717E-04	17.5934E-03

A 7 COEFFICIENTS

-14.2159E-04	-14.1114E-04
11.5912E-03	13.9712E-03

B 7 COEFFICIENTS

13.3851E-04	20.5031E-04
78.3281E-04	10.0977E-03

A 8 COEFFICIENTS

-13.0515E-05	-33.5245E-06
12.7509E-04	14.7852E-04

B 8 COEFFICIENTS

16.3824E-05	24.6929E-05
22.2372E-05	-82.8043E-05

A 9 COEFFICIENTS

-46.5816E-06	-11.0951E-05
46.4886E-05	19.8482E-04

B 9 COEFFICIENTS

88.2182E-06	44.4661E-05
70.3954E-05	20.3948E-04

A10 COEFFICIENTS

13.0906E-05	18.3499E-05
-20.0284E-05	25.8491E-05

B10 COEFFICIENTS

90.3189E-06	15.5450E-05
-83.1601E-05	-19.4652E-04

A11 COEFFICIENTS

48.0838E-05	46.4416E-05
-39.1808E-04	-42.6440E-04

B11 COEFFICIENTS

44.1368E-05	71.6457E-05
25.4166E-04	34.3360E-04

A12 COEFFICIENTS

69.3890E-05	10.7174E-04
-11.7335E-03	-56.8159E-04

B12 COEFFICIENTS

18.1969E-04	10.4698E-04
72.4322E-05	49.4734E-04

A13 COEFFICIENTS

-10.5981E-04	21.8299E-04
-31.3044E-03	-25.1487E-03

B13 COEFFICIENTS

65.3715E-04	49.9286E-04
-15.5567E-03	11.1807E-03

A14 COEFFICIENTS

-14.0456E-03	41.5557E-05
-11.2766E-03	-12.5759E-03

B14 COEFFICIENTS

82.7694E-04	31.7815E-04
-69.1282E-03	14.8758E-04

A15 COEFFICIENTS

-44.4240E-03	-32.8994E-03
95.5889E-03	-29.9151E-03

B15 COEFFICIENTS

-89.5944E-04	89.1404E-04
-11.9456E-02	-98.0347E-03

A16 COEFFICIENTS

-58.8642E-03	-89.9490E-03
25.1095E-02	-73.5972E-03

B16 COEFFICIENTS

-79.9327E-03	32.9036E-03
61.6862E-04	-17.7106E-02

A17 COEFFICIENTS

-72.2501E-04	-88.8079E-03
12.0953E-02	-16.6585E-02

B17 COEFFICIENTS

-10.9792E-02	15.8754E-02
16.2914E-02	-75.1278E-03

A18 COEFFICIENTS

10.7442E-02	18.1045E-02
-16.6533E-02	15.5927E-03

B18 COEFFICIENTS

-48.8196E-11	-91.2150E-12
94.8963E-11	-90.8258E-11

INCREMENT OF RUN EQUALS 36 STEPS

FOR THIS RUN;
B= .97 MU= 2.40 GAMMA= 8.00 PSQUARE= 1.00

POINCARE EXPONENTS

OMEGA1=-20.36572E-01 OMEGA2= 95.58700E-03

SYSTEM OF EIGENVECTORS

20.5712E-02	43.2509E-02
-97.8613E-02	90.1630E-02

A 0 COEFFICIENTS

11.8772E-01	10.2515E-01
86.2073E-02	13.6359E-01

B 0 COEFFICIENTS

.0000E+00	.0000E+00
.0000E+00	.0000E+00

A 1 COEFFICIENTS

-21.5499E-02	46.5291E-02
19.6603E-02	11.3453E-01

B 1 COEFFICIENTS

-28.9105E-02	13.4273E-02
86.4842E-03	-21.5431E-02

A 2 COEFFICIENTS

14.8039E-02	84.5943E-02
95.8541E-02	70.8016E-02

B 2 COEFFICIENTS

42.7057E-03	36.8620E-03
39.8561E-02	-23.6625E-02

A 3 COEFFICIENTS

23.1093E-02	68.2755E-02
57.5519E-02	80.9007E-02

B 3 COEFFICIENTS

78.6651E-04	-89.6951E-03
-20.2031E-02	-58.3766E-02

A 4 COEFFICIENTS

81.0408E-03	42.7697E-02
59.7315E-03	-13.5355E-02

B 4 COEFFICIENTS

-22.0729E-02	-29.1448E-02
-40.0155E-02	-17.2588E-02

A 5 COEFFICIENTS

10.8282E-02	39.9897E-02
10.8934E-02	37.3628E-02

B 5 COEFFICIENTS

84.7035E-03	15.2980E-02
75.8592E-02	20.3044E-02

A 6 COEFFICIENTS

27.9636E-02	49.7259E-02
73.8246E-02	50.1511E-02

B 6 COEFFICIENTS

-74.8492E-02	-67.5640E-02
-33.8013E-04	-15.4892E-02

A 7 COEFFICIENTS

-68.7952E-02	-56.6376E-02
-48.5174E-03	-26.1896E-02

B 7 COEFFICIENTS

41.0381E-02	57.0929E-02
65.7802E-02	44.0293E-02

A 8 COEFFICIENTS

71.6266E-03	18.5705E-02
41.6068E-02	17.5057E-02

B 8 COEFFICIENTS

13.9513E-02	35.3739E-02
15.3285E-02	56.4575E-02

A 9 COEFFICIENTS

45.0195E-03	85.8710E-03
-48.6608E-03	-12.3636E-03

B 9 COEFFICIENTS

84.3689E-04	29.6658E-02
47.7471E-02	17.1040E-02

A10 COEFFICIENTS

-48.3488E-02	-51.1885E-02
-31.0404E-02	-35.9755E-02

B10 COEFFICIENTS

23.2088E-04	31.3986E-02
71.4163E-02	92.1507E-02

A11 COEFFICIENTS

23.0722E-03	21.5094E-03
80.4364E-02	37.7143E-02

B11 COEFFICIENTS

34.8169E-02	72.0564E-02
72.0456E-02	59.0006E-02

A12 COEFFICIENTS

-35.2420E-02	-38.6831E-02
12.2056E-02	77.5380E-03

B12 COEFFICIENTS

-26.8505E-03	38.7129E-02
-19.1759E-02	34.9878E-02

A13 COEFFICIENTS

12.8885E-02	80.7772E-03
-11.9757E-02	-97.0911E-03

B13 COEFFICIENTS

10.1510E-01	14.9400E-01
92.2498E-02	10.7946E-01

A14 COEFFICIENTS

20.6526E-02	16.5659E-02
-13.7431E-02	33.9748E-02

B14 COEFFICIENTS

-81.9684E-03	30.7876E-02
10.6607E-02	43.6482E-02

A15 COEFFICIENTS

-57.4714E-03	-11.5097E-02
45.0695E-03	-17.3672E-02

B15 COEFFICIENTS

78.3321E-03	46.9790E-02
45.4000E-02	22.6939E-02

A16 COEFFICIENTS

-26.1592E-02	-42.3541E-02
-32.6501E-02	-55.6018E-03

B16 COEFFICIENTS

54.4707E-02	94.7728E-02
47.6885E-02	58.0904E-02

A17 COEFFICIENTS

63.2308E-02	51.8229E-02
16.9784E-03	-13.0551E-02

B17 COEFFICIENTS

36.0755E-03	49.6161E-02
-41.7061E-02	-19.7202E-02

A18 COEFFICIENTS

-77.5633E-03	74.5561E-03
-99.7068E-02	-45.6649E-02

B18 COEFFICIENTS

54.3097E-03	32.7916E-02
20.8531E-02	20.1257E-02

INCREMENT OF RUN EQUALS 360 STEPS

FOR THIS RUN;
B= .97 MU= 2.40 GAMMA=10.00 PSQUARE= 1.00

POINCARE EXPONENTS

OMEGA1=-25.12578E-01 OMEGA2= 81.90400E-03

SYSTEM OF EIGENVECTORS

-18.3184E-02	50.7958E-02
98.3079E-02	86.1382E-02

A 0 COEFFICIENTS

33.9228E-02	-56.6455E-02
-83.4099E-02	-46.3378E-03

B 0 COEFFICIENTS

.0000E+00	.0000E+00
.0000E+00	.0000E+00

A 1 COEFFICIENTS

94.0188E-03	24.6571E-02
89.0624E-03	54.6691E-02

B 1 COEFFICIENTS

30.7057E-02	52.6371E-02
-86.5526E-02	-20.3481E-02

A 2 COEFFICIENTS

-15.6284E-02	37.4191E-02
93.9369E-02	15.6794E-02

B 2 COEFFICIENTS

26.4231E-02	63.0430E-03
-35.1361E-02	-74.3271E-02

A 3 COEFFICIENTS

-18.2564E-02	15.2363E-02
63.5598E-02	-83.6715E-03

B 3 COEFFICIENTS

52.8993E-03	-32.1217E-03
41.4744E-02	-45.9820E-02

A 4 COEFFICIENTS

-86.1619E-03	16.1701E-03
70.9883E-03	16.2254E-05

B 4 COEFFICIENTS

-40.9324E-03	-29.5246E-05
44.7398E-02	-65.0169E-03

A 5 COEFFICIENTS

-14.3148E-03	19.1124E-04
-15.7159E-02	13.3819E-02

B 5 COEFFICIENTS

-42.2562E-03	26.6900E-03
17.7584E-02	-76.3040E-04

A 6 COEFFICIENTS

56.1112E-04	43.8628E-05
-10.8990E-02	65.9821E-03

B 6 COEFFICIENTS

-18.8645E-03	11.0360E-03
13.4928E-03	-27.2000E-04

A 7 COEFFICIENTS

58.0747E-04	-10.2628E-04
-40.2135E-03	41.8699E-03

B 7 COEFFICIENTS

-62.4839E-04	60.7098E-04
-25.5105E-03	73.7344E-04

A 8 COEFFICIENTS

24.4285E-04	-20.3257E-06
-75.5173E-04	11.7410E-03

B 8 COEFFICIENTS

-24.4200E-04	16.5948E-04
-14.4111E-03	-15.5089E-04

A 9 COEFFICIENTS

88.2273E-05	-15.8688E-05
-34.5439E-05	51.8213E-04

B 9 COEFFICIENTS

-17.4481E-04	11.5727E-04
-77.7139E-04	29.9064E-04

A10 COEFFICIENTS

-14.9817E-05	29.3766E-05
22.1497E-04	-16.2925E-04

B10 COEFFICIENTS

-18.8793E-04	77.8146E-05
-59.4440E-04	-32.7850E-04

A11 COEFFICIENTS

-13.6594E-04	39.0955E-05
13.1050E-03	-13.1224E-03

B11 COEFFICIENTS

-29.9311E-04	20.9393E-04
-93.0120E-04	19.5546E-04

A12 COEFFICIENTS

-13.7497E-04	52.9246E-07
35.7185E-03	-21.5120E-03

B12 COEFFICIENTS

-70.7485E-04	37.1427E-04
40.8066E-04	-26.4279E-04

A13 COEFFICIENTS

52.3338E-04	-59.9181E-05
52.0155E-03	-45.0293E-03

B13 COEFFICIENTS

-14.7054E-03	88.5219E-04
58.7443E-03	-30.3782E-04

A14 COEFFICIENTS

29.1538E-03	-53.4949E-04
-24.1669E-03	-86.0706E-06

B14 COEFFICIENTS

-14.1407E-03	-10.5155E-05
14.8762E-02	-22.2716E-03

A15 COEFFICIENTS

61.2629E-03	-50.7777E-03
-21.2259E-02	27.4485E-03

B15 COEFFICIENTS

17.2748E-03	-10.7488E-03
13.7989E-02	-15.3485E-02

A16 COEFFICIENTS

52.4931E-03	-12.4739E-02
-31.3567E-02	-52.3904E-03

B16 COEFFICIENTS

87.8363E-03	21.0055E-03
-11.7333E-02	-24.7853E-02

A17 COEFFICIENTS

-30.9550E-03	-82.1938E-03
-30.1174E-03	-18.2483E-02

B17 COEFFICIENTS

10.2235E-02	17.5442E-02
-28.8593E-02	-67.8751E-03

A18 COEFFICIENTS

-11.2689E-02	18.8797E-02
27.7605E-02	15.3229E-03

B18 COEFFICIENTS

47.2773E-11	-25.5162E-11
-17.2389E-10	-89.0636E-11

INCREMENT OF RUN EQUALS 360 STEPS

FOR THIS RUN;
B= .97 MU= 2.40 GAMMA=12.00 PSQUARE= 1.00

POINCARE EXPONENTS

OMEGA1=-29.86568E-01 OMEGA2= 56.85900E-03

SYSTEM OF EIGENVECTORS

-15.1443E-02	57.8038E-02
98.8466E-02	81.6010E-02

A 0 COEFFICIENTS

15.0218E-02	-59.1439E-02
-31.2415E-01	-33.5516E-03

B 0 COEFFICIENTS

.0000E+00	.0000E+00
.0000E+00	.0000E+00

A 1 COEFFICIENTS

-16.6921E-02	24.4537E-02
-19.8158E-01	58.0557E-02

B 1 COEFFICIENTS

19.4244E-02	56.6499E-02
-41.1836E-02	-21.2354E-02

A 2 COEFFICIENTS

-59.8728E-02	42.3082E-02
-42.7799E-02	13.4728E-02

B 2 COEFFICIENTS

22.7456E-02	55.2921E-03
52.2407E-02	-84.3089E-02

A 3 COEFFICIENTS

-70.4334E-02	17.2261E-02
-40.3020E-03	-17.7768E-02

B 3 COEFFICIENTS

17.3728E-02	-62.6099E-03
16.0456E-01	-52.0484E-02

A 4 COEFFICIENTS

-54.9770E-02	21.2675E-03
-29.6615E-02	-36.9130E-03

B 4 COEFFICIENTS

17.9551E-02	-95.4169E-04
16.8450E-01	-86.0159E-03

A 5 COEFFICIENTS

-36.9618E-02	70.0140E-04
-43.9165E-02	14.9868E-02

B 5 COEFFICIENTS

22.0589E-02	29.8431E-03
12.2973E-01	-33.7015E-03

A 6 COEFFICIENTS

-25.8603E-02	40.7577E-04
-32.7618E-02	86.8063E-03

B 6 COEFFICIENTS

25.5376E-02	14.4908E-03
85.8603E-02	-24.8821E-03

A 7 COEFFICIENTS

-19.5167E-02	27.9879E-05
-17.5949E-02	57.8227E-03

B 7 COEFFICIENTS

26.5199E-02	83.7907E-04
68.7266E-02	-19.8926E-04

A 8 COEFFICIENTS

-15.2316E-02	36.1309E-05
-73.5266E-03	19.6705E-03

B 8 COEFFICIENTS

25.8109E-02	27.7880E-04
62.3278E-02	-50.0961E-04

A 9 COEFFICIENTS

-11.6478E-02	-12.6293E-05
-15.6179E-03	80.4737E-04

B 9 COEFFICIENTS

24.4503E-02	17.9440E-04
57.9197E-02	24.3742E-04

A10 COEFFICIENTS

-84.6283E-03	23.3536E-05
27.0108E-03	-27.4632E-04

B10 COEFFICIENTS

22.8905E-02	13.0433E-04
53.9185E-02	-50.7268E-04

A11 COEFFICIENTS

-54.6697E-03	-13.1712E-06
76.4848E-03	-17.9722E-03

B11 COEFFICIENTS

21.1512E-02	29.0471E-04
50.6659E-02	-14.8924E-04

A12 COEFFICIENTS

-21.4463E-03	-11.6750E-04
13.7058E-02	-28.2981E-03

B12 COEFFICIENTS

18.9063E-02	48.7739E-04
51.4276E-02	-10.4828E-03

A13 COEFFICIENTS

24.7748E-03	-22.7674E-04
18.1160E-02	-50.4803E-03

B13 COEFFICIENTS

15.8992E-02	98.9234E-04
59.2652E-02	-11.9699E-03

A14 COEFFICIENTS

91.7290E-03	-70.3685E-04
13.7644E-02	12.2064E-03

B14 COEFFICIENTS

12.7416E-02	-31.9444E-04
70.2855E-02	-29.4447E-03

A15 COEFFICIENTS

14.8245E-02	-57.4064E-03
54.6199E-03	58.6905E-03

B15 COEFFICIENTS

10.8113E-02	-20.9230E-03
63.8075E-02	-17.3790E-02

A16 COEFFICIENTS

11.6430E-02	-14.1038E-02
18.4950E-02	-45.1034E-03

B16 COEFFICIENTS

10.9054E-02	18.4163E-03
24.1557E-02	-28.1156E-02

A17 COEFFICIENTS

-25.5512E-03	-81.5185E-03
70.3416E-02	-19.3843E-02

B17 COEFFICIENTS

81.2972E-03	18.8814E-02
-10.3954E-02	-70.8469E-03

A18 COEFFICIENTS

-13.0619E-02	19.7119E-02
10.8443E-01	11.0081E-03

B18 COEFFICIENTS

70.5010E-11	-32.0229E-11
11.2050E-10	-91.3379E-11

INCREMENT OF RUN EQUALS 360 STEPS

FOR THIS RUN;
B= .97 MU= 2.40 GAMMA=14.00 PSQUARE= 1.00

POINCARE EXPONENTS

OMEGA1=-30.31760E-01 OMEGA2= 25.30000E-03

SYSTEM OF EIGENVECTORS

-11.3653E-02	64.3938E-02
99.3521E-02	76.5077E-02

A 0 COEFFICIENTS

-28.6256E-02	-62.4677E-02
-54.5366E-01	-15.6881E-03

B 0 COEFFICIENTS

.0000E+00	.0000E+00
.0000E+00	.0000E+00

A 1 COEFFICIENTS

-62.0704E-02	24.4110E-02
-44.7871E-01	61.9486E-02

B 1 COEFFICIENTS

-41.6308E-03	61.3108E-02
75.1732E-02	-22.8636E-02

A 2 COEFFICIENTS

-11.6490E-01	47.2723E-02
-25.6014E-01	11.6321E-02

B 2 COEFFICIENTS

10.9609E-02	52.1135E-03
20.1084E-01	-94.4216E-02

A 3 COEFFICIENTS

-13.3823E-01	19.0480E-02
-11.8021E-01	-27.3447E-02

B 3 COEFFICIENTS

35.3525E-02	-92.8687E-03
29.7245E-01	-57.3983E-02

A 4 COEFFICIENTS

-11.4031E-01	24.1271E-03
-68.2079E-02	-79.7181E-03

B 4 COEFFICIENTS

53.5045E-02	-20.1034E-03
29.9723E-01	-97.5216E-03

A 5 COEFFICIENTS

-85.0163E-02	11.6965E-03
-57.2038E-02	15.9057E-02

B 5 COEFFICIENTS

62.1112E-02	31.6922E-03
24.7227E-01	-58.2183E-03

A 6 COEFFICIENTS

-62.3884E-02	84.0106E-04
-44.8369E-02	10.3111E-02

B 6 COEFFICIENTS

64.7382E-02	17.2363E-03
19.5639E-01	-51.5006E-03

A 7 COEFFICIENTS

-47.0747E-02	22.6032E-04
-27.2968E-02	72.4014E-03

B 7 COEFFICIENTS

64.0689E-02	10.5232E-03
16.3363E-01	-16.3190E-03

A 8 COEFFICIENTS

-36.0972E-02	10.8893E-04
-11.2331E-02	28.1818E-03

B 8 COEFFICIENTS

61.5902E-02	40.2642E-04
14.5145E-01	-11.5496E-03

A 9 COEFFICIENTS

-27.2806E-02	-18.2606E-07
88.1660E-04	11.6647E-03

B 9 COEFFICIENTS

58.2695E-02	25.9673E-04
13.2970E-01	18.7921E-05

A10 COEFFICIENTS

-19.5600E-02	93.3743E-06
10.4976E-02	-34.0530E-04

B10 COEFFICIENTS

54.6226E-02	19.3802E-04
12.3589E-01	-82.5379E-04

A11 COEFFICIENTS

-12.1975E-02	-63.3059E-05
19.3931E-02	-22.0698E-03

B11 COEFFICIENTS

50.7416E-02	36.8685E-04
11.7277E-01	-66.8696E-04

A12 COEFFICIENTS

-42.3835E-03	-25.6644E-04
27.4877E-02	-33.4701E-03

B12 COEFFICIENTS

46.3925E-02	58.1242E-04
11.6697E-01	-19.8441E-03

A13 COEFFICIENTS

54.8264E-03	-38.2018E-04
32.9964E-02	-53.6207E-03

B13 COEFFICIENTS

41.1004E-02	10.4990E-03
12.3573E-01	-20.4115E-03

A14 COEFFICIENTS

16.7868E-02	-79.7684E-04
37.4438E-02	26.4079E-03

B14 COEFFICIENTS

33.9602E-02	-67.2289E-04
13.1674E-01	-33.4724E-03

A15 COEFFICIENTS

24.5662E-02	-63.4748E-03
54.4406E-02	90.4510E-03

B15 COEFFICIENTS

23.7635E-02	-31.0220E-03
12.2218E-01	-19.1721E-02

A16 COEFFICIENTS

19.5892E-02	-15.7587E-02
10.0601E-01	-39.0458E-03

B16 COEFFICIENTS

11.5841E-02	17.3503E-03
82.1200E-02	-31.4901E-02

A17 COEFFICIENTS

19.1216E-03	-81.3795E-03
16.4605E-01	-20.6899E-02

B17 COEFFICIENTS

25.6071E-03	20.4346E-02
32.5043E-02	-76.2910E-03

A18 COEFFICIENTS

-90.8445E-03	20.8192E-02
19.7112E-01	49.9261E-04

B18 COEFFICIENTS

91.6327E-11	-38.2674E-11
58.7632E-10	-95.2217E-11

INCREMENT OF RUN EQUALS 360 STEPS

FOR THIS RUN;
B= .97 MU= 2.40 GAMMA=16.00 PSQUARE= 1.00

POINCARE EXPONENTS

OMEGA1=-33.36009E-01 OMEGA2=-10.56400E-03

SYSTEM OF EIGENVECTORS

-70.7829E-03	70.5935E-02
99.7492E-02	70.8276E-02

A 0 COEFFICIENTS

-32.7740E-01	-66.3098E-02
-22.4760E+00	71.4889E-04

B 0 COEFFICIENTS

.0000E+00	.0000E+00
.0000E+00	.0000E+00

A 1 COEFFICIENTS

-41.6939E-01	24.3642E-02
-20.0876E+00	66.1153E-02

B 1 COEFFICIENTS

-60.2491E-02	66.3488E-02
62.0587E-01	-25.0699E-02

A 2 COEFFICIENTS

-56.4746E-01	52.2781E-02
-14.3530E+00	10.0159E-02

B 2 COEFFICIENTS

77.6956E-03	52.7564E-03
11.1085E+00	-10.4623E-01

A 3 COEFFICIENTS

-60.5523E-01	20.7920E-02
-84.1430E-01	-36.9315E-02

B 3 COEFFICIENTS

15.5215E-01	-12.2506E-02
13.1475E+00	-62.2716E-02

A 4 COEFFICIENTS

-53.0780E-01	25.2434E-03
-45.0764E-01	-12.5413E-02

B 4 COEFFICIENTS

27.3656E-01	-31.3130E-03
12.4135E+00	-10.1257E-02

A 5 COEFFICIENTS

-41.9517E-01	15.7361E-03
-26.0204E-01	16.3169E-02

B 5 COEFFICIENTS

32.8337E-01	32.6071E-03
10.5793E+00	-79.7277E-03

A 6 COEFFICIENTS

-32.2891E-01	13.0426E-03
-16.2615E-01	11.4796E-02

B 6 COEFFICIENTS

34.0446E-01	19.2846E-03
89.4790E-01	-80.3061E-03

A 7 COEFFICIENTS

-24.9460E-01	47.5139E-04
-90.8492E-02	84.3722E-03

B 7 COEFFICIENTS

33.3579E-01	12.3493E-03
78.1620E-01	-34.5395E-03

A 8 COEFFICIENTS

-19.2706E-01	21.4708E-04
-29.1518E-02	36.2859E-03

B 8 COEFFICIENTS

31.9267E-01	52.8384E-04
70.3433E-01	-21.2346E-03

A 9 COEFFICIENTS

-14.6052E-01	24.1306E-05
22.8217E-02	15.7427E-03

B 9 COEFFICIENTS

30.2022E-01	35.0538E-04
64.4257E-01	-42.6224E-04

A10 COEFFICIENTS

-10.5143E-01	-97.6937E-06
66.6679E-02	-33.2802E-04

B10 COEFFICIENTS

28.3583E-01	26.3691E-04
59.7441E-01	-13.0927E-03

A11 COEFFICIENTS

-66.6088E-02	-14.0386E-04
10.5208E-01	-24.9478E-03

B11 COEFFICIENTS

26.4281E-01	43.9581E-04
56.3059E-01	-13.3980E-03

A12 COEFFICIENTS

-26.9831E-02	-40.6537E-04
14.0762E-01	-36.9449E-03

B12 COEFFICIENTS

24.3018E-01	65.2893E-04
54.5844E-01	-30.0009E-03

A13 COEFFICIENTS

16.8220E-02	-51.4264E-04
18.0370E-01	-55.0155E-03

B13 COEFFICIENTS

21.6091E-01	10.7987E-03
55.0376E-01	-27.8827E-03

A14 COEFFICIENTS

62.6280E-02	-83.3457E-04
24.7804E-01	41.5842E-03

B14 COEFFICIENTS

17.5631E-01	-10.4663E-03
56.6184E-01	-34.9271E-03

A15 COEFFICIENTS

93.8738E-02	-69.2828E-03
37.9909E-01	12.2272E-02

B15 COEFFICIENTS

11.4500E-01	-40.9137E-03
54.9105E-01	-20.8076E-02

A16 COEFFICIENTS

84.5749E-02	-17.4275E-02
57.8587E-01	-33.7430E-03

B16 COEFFICIENTS

44.1633E-02	17.5578E-03
44.2523E-01	-34.8948E-02

A17 COEFFICIENTS

37.7932E-02	-81.2268E-03
76.9926E-01	-22.0874E-02

B17 COEFFICIENTS

62.2276E-04	22.1134E-02
24.2478E-01	-83.6653E-03

A18 COEFFICIENTS

88.7491E-03	22.0992E-02
84.9560E-01	-26.8703E-04

B18 COEFFICIENTS

42.9164E-10	-44.3990E-11
31.2739E-09	-10.0058E-10

Appendix D.

Mode Controllability Matrices Element Fourier Series Coefficients

INCREMENT OF RUN EQUALS 360 STEPS

FOR THIS RUN:
B= .97 MU= 2.40 GAMMA= 2.00 PSQUARE= 1.00

POINCARE EXPONENTS

OMEGA1=-42.54970E-02 OMEGA2=-62.33300E-03

SYSTEM OF EIGENVECTORS

16.5818E-02	31.1379E-02
-98.6156E-02	95.0286E-02

A 0 COEFFICIENTS

17.2936E-01 -67.1089E-02

B 0 COEFFICIENTS

.0000E+00 .0000E+00

A 1 COEFFICIENTS

-13.5909E-01 64.3571E-02

B 1 COEFFICIENTS

-15.1100E-01 -15.2794E-01

A 2 COEFFICIENTS

-98.0245E-03 80.0865E-03

B 2 COEFFICIENTS

-29.8593E-02 -35.4408E-02

A 3 COEFFICIENTS

-12.5109E-02 22.6213E-04

B 3 COEFFICIENTS

52.5992E-04 -14.1199E-03

A 4 COEFFICIENTS

44.7952E-03	-32.3929E-03

B 4 COEFFICIENTS

38.6335E-03	28.9450E-03

A 5 COEFFICIENTS

-71.6853E-04	-60.4914E-05

B 5 COEFFICIENTS

18.0087E-04	91.9358E-04

A 6 COEFFICIENTS

88.9999E-04	-29.9810E-04

B 6 COEFFICIENTS

58.9212E-04	36.3121E-04

A 7 COEFFICIENTS

-15.1086E-04	10.4907E-04

B 7 COEFFICIENTS

-29.1683E-04	23.4895E-04

A 8 COEFFICIENTS

16.6153E-04	-35.3448E-05

B 8 COEFFICIENTS

25.3774E-04	65.4536E-05

A 9 COEFFICIENTS

13.5823E-05	35.6150E-05

B 9 COEFFICIENTS

-20.4539E-04	12.3178E-04

A10 COEFFICIENTS

-47.9126E-05	30.7768E-05

B10 COEFFICIENTS

16.2466E-04	28.2642E-05

A11 COEFFICIENTS

89.6143E-05	-12.7433E-05

B11 COEFFICIENTS

-15.8331E-04	10.7334E-04

A12 COEFFICIENTS

-31.8666E-04	12.0797E-04

B12 COEFFICIENTS

21.6224E-04	11.5410E-04

A13 COEFFICIENTS

26.6464E-04	28.4903E-05

B13 COEFFICIENTS

41.0272E-05	31.1129E-04

A14 COEFFICIENTS

-15.1151E-03	10.9167E-03

B14 COEFFICIENTS

12.8465E-03	95.8140E-04

A15 COEFFICIENTS

41.8141E-03	-74.6242E-05

B15 COEFFICIENTS

17.3734E-04	-47.3791E-04

A16 COEFFICIENTS

32.5848E-03 -26.6601E-03

B16 COEFFICIENTS

-99.5927E-03 -11.8176E-02

A17 COEFFICIENTS

45.3038E-02 -21.4549E-02

B17 COEFFICIENTS

-50.3656E-02 -50.9333E-02

A18 COEFFICIENTS

-57.6503E-02 22.3700E-02

B18 COEFFICIENTS

-21.6881E-11 -88.3967E-11

INCREMENT OF RUN EQUALS 360 STEPS

FOR THIS RUN;
B= .97 MU= 2.40 GAMMA= 4.00 PSQUARE= 1.00

POINCARE EXPONENTS

OMEGA1=-10.09043E-01 OMEGA2= 33.37400E-03

SYSTEM OF EIGENVECTORS

19.5071E-02	27.3698E-02
-98.0789E-02	96.1816E-02

A 0 COEFFICIENTS

30.1820E-01 -13.7756E-01

B 0 COEFFICIENTS

.0000E+00 .0000E+00

A 1 COEFFICIENTS

-15.1648E-01 78.6900E-02

B 1 COEFFICIENTS

-17.1629E-01 -14.0303E-01

A 2 COEFFICIENTS

-32.9207E-02 34.7404E-02

B 2 COEFFICIENTS

-38.3547E-02 -42.5539E-02

A 3 COEFFICIENTS

-38.9096E-02 75.5412E-03

B 3 COEFFICIENTS

94.6384E-03 -26.8808E-05

A 4 COEFFICIENTS

10.9579E-02 -80.3465E-03

B 4 COEFFICIENTS

89.6815E-03 28.3663E-03

A 5 COEFFICIENTS

-14.1635E-03 -81.2760E-04

B 5 COEFFICIENTS

12.0177E-03 18.6757E-03

A 6 COEFFICIENTS

29.3646E-03 -98.1474E-04

B 6 COEFFICIENTS

96.1625E-04 51.9469E-04

A 7 COEFFICIENTS

-56.6828E-04 24.7467E-04

B 7 COEFFICIENTS

-75.5487E-04 63.9953E-04

A 8 COEFFICIENTS

52.7498E-04 -10.3643E-04

B 8 COEFFICIENTS

56.9981E-04 46.8455E-05

A 9 COEFFICIENTS

-24.7636E-05 67.0423E-05

B 9 COEFFICIENTS

-53.9079E-04 32.3870E-04

A10 COEFFICIENTS

-12.1646E-04 77.5763E-05

B10 COEFFICIENTS

39.6596E-04 18.0515E-06

A11 COEFFICIENTS

26.4455E-04 -51.1377E-05

B11 COEFFICIENTS

-42.3032E-04 29.1320E-04

A12 COEFFICIENTS

-10.2502E-03 37.9081E-04

B12 COEFFICIENTS

38.4164E-04 15.5028E-04

A13 COEFFICIENTS

53.7528E-04 27.8451E-04

B13 COEFFICIENTS

34.0985E-04 63.6847E-04

A14 COEFFICIENTS

-36.9928E-03 27.0890E-03

B14 COEFFICIENTS

29.8939E-03 93.3017E-04

A15 COEFFICIENTS

12.9997E-02 -25.2018E-03

B15 COEFFICIENTS

31.4443E-03 -15.7158E-05

A16 COEFFICIENTS

10.9472E-02	-11.5703E-02

B16 COEFFICIENTS

-12.7973E-02	-14.1906E-02

A17 COEFFICIENTS

50.5546E-02	-26.2355E-02

B17 COEFFICIENTS

-57.2088E-02	-46.7724E-02

A18 COEFFICIENTS

-10.0623E-01	45.9204E-02

B18 COEFFICIENTS

39.6555E-11	-12.1956E-10

INCREMENT OF RUN EQUALS 360 STEPS

FOR THIS RUN;
B= .97 MU= 2.40 GAMMA= 6.00 PSQUARE= 1.00

POINCARE EXPONENTS

OMEGA1=-15.49838E-01 OMEGA2= 86.36900E-03

SYSTEM OF EIGENVECTORS

21.2337E-02	35.1497E-02
-97.7197E-02	93.6189E-02

A 0 COEFFICIENTS

39.1804E-01 -15.8224E-01

B 0 COEFFICIENTS

.0000E+00 .0000E+00

A 1 COEFFICIENTS

-17.9501E-01 68.9259E-02

B 1 COEFFICIENTS

-19.2140E-01 -12.9562E-01

A 2 COEFFICIENTS

-40.7326E-02 49.9553E-02

B 2 COEFFICIENTS

-32.4624E-02 -31.4557E-02

A 3 COEFFICIENTS

-65.9545E-02 12.2059E-02

B 3 COEFFICIENTS

28.1964E-02 47.9613E-03

A 4 COEFFICIENTS

22.0567E-02	-10.2411E-02

B 4 COEFFICIENTS

14.1571E-02	18.4471E-03

A 5 COEFFICIENTS

-22.4732E-03	-16.1852E-03

B 5 COEFFICIENTS

21.0586E-03	15.6389E-03

A 6 COEFFICIENTS

61.9723E-03	-15.3209E-03

B 6 COEFFICIENTS

25.3041E-04	44.1703E-04

A 7 COEFFICIENTS

-15.0067E-03	29.9798E-04

B 7 COEFFICIENTS

-15.1755E-03	88.1646E-04

A 8 COEFFICIENTS

10.6436E-03	-15.7823E-04

B 8 COEFFICIENTS

78.9533E-04	38.8224E-05

A 9 COEFFICIENTS

-19.3759E-04	88.2676E-05

B 9 COEFFICIENTS

-10.2404E-03	43.8635E-04

A10 COEFFICIENTS

-22.2283E-04   10.1253E-04

B10 COEFFICIENTS

63.1807E-04   -16.6157E-05

A11 COEFFICIENTS

56.3769E-04   -63.4332E-05

B11 COEFFICIENTS

-84.3872E-04   40.1213E-04

A12 COEFFICIENTS

-21.4691E-03   57.6457E-04

B12 COEFFICIENTS

22.4074E-04   12.0130E-04

A13 COEFFICIENTS

84.5376E-04   54.5814E-04

B13 COEFFICIENTS

57.9542E-04   54.3429E-04

A14 COEFFICIENTS

-74.4237E-03   34.5410E-03

B14 COEFFICIENTS

47.4022E-03   59.8488E-04

A15 COEFFICIENTS

22.0386E-02   -40.7304E-03

B15 COEFFICIENTS

93.7542E-03   15.9163E-03

A16 COEFFICIENTS

13.5244E-02	-16.6377E-02

B16 COEFFICIENTS

-10.8323E-02	-10.4925E-02

A17 COEFFICIENTS

59.8508E-02	-22.9828E-02

B17 COEFFICIENTS

-64.0461E-02	-43.1931E-02

A18 COEFFICIENTS

-13.0636E-01	52.7451E-02

B18 COEFFICIENTS

12.8949E-10	-12.5211E-10

INCREMENT OF RUN EQUALS 36 STEPS

FOR THIS RUN:
B= .97 MU= 2.40 GAMMA= 8.00 PSQUARE= 1.00

POINCARE EXPONENTS

OMEGA1=-20.36572E-01 OMEGA2= 95.58700E-03

SYSTEM OF EIGENVECTORS

20.5712E-02	43.2509E-02
-97.8613E-02	90.1630E-02

A 0 COEFFICIENTS

38.3032E-01 -83.5483E-01

B 0 COEFFICIENTS

.0000E+00 .0000E+00

A 1 COEFFICIENTS

19.4091E-03 41.5386E-01

B 1 COEFFICIENTS

18.0379E-02 14.8895E-01

A 2 COEFFICIENTS

-53.2786E-02 15.6524E-01

B 2 COEFFICIENTS

44.7947E-01 -44.5186E-01

A 3 COEFFICIENTS

72.1959E-01 -56.2319E-01

B 3 COEFFICIENTS

-39.0440E-01 20.1968E-01

A 4 COEFFICIENTS

-58.8626E-01 55.8466E-01

B 4 COEFFICIENTS

-21.0365E-01 61.3445E-02

A 5 COEFFICIENTS

17.5566E-01 -16.0705E-01

B 5 COEFFICIENTS

25.0245E-01 -32.1868E-01

A 6 COEFFICIENTS

46.7009E-01 -51.0982E-01

B 6 COEFFICIENTS

-48.7106E-01 40.2055E-01

A 7 COEFFICIENTS

-60.2631E-01 51.2633E-01

B 7 COEFFICIENTS

29.5155E-01 -33.3190E-01

A 8 COEFFICIENTS

21.1470E-01 -28.6282E-01

B 8 COEFFICIENTS

31.5881E-01 -31.3069E-01

A 9 COEFFICIENTS

-32.3638E-02 -74.7150E-02

B 9 COEFFICIENTS

-45.7714E-01 47.4797E-01

A10 COEFFICIENTS

-62.3159E-01	48.8875E-01

B10 COEFFICIENTS

49.0673E-01	-42.8498E-01

A11 COEFFICIENTS

40.1700E-01	-45.3392E-01

B11 COEFFICIENTS

15.7928E-02	50.1239E-02

A12 COEFFICIENTS

-10.2011E-01	67.6856E-02

B12 COEFFICIENTS

-32.1101E-01	42.0889E-01

A13 COEFFICIENTS

-26.8316E-01	22.2119E-01

B13 COEFFICIENTS

84.2448E-01	-68.6305E-01

A14 COEFFICIENTS

36.6386E-01	-33.4362E-01

B14 COEFFICIENTS

-11.6566E-01	19.4679E-01

A15 COEFFICIENTS

-42.9179E-01	43.0188E-01

B15 COEFFICIENTS

-17.7343E-01	23.3938E-01

A16 COEFFICIENTS

-19.5160E-02	59.6381E-02

B16 COEFFICIENTS

66.1177E-01	-55.6346E-01

A17 COEFFICIENTS

26.1155E-01	-27.9550E-01

B17 COEFFICIENTS

-49.3272E-01	54.6537E-01

A18 COEFFICIENTS

-90.5831E-02	38.2448E-01

B18 COEFFICIENTS

-73.6977E-03	-72.2681E-04

INCREMENT OF RUN EQUALS 360 STEPS

FOR THIS RUN;
B= .97 MU= 2.40 GAMMA=10.00 PSQUARE= 1.00

POINCARE EXPONENTS

OMEGA1=-25.12578E-01 OMEGA2= 81.90400E-03

SYSTEM OF EIGENVECTORS

-18.3184E-02	50.7958E-02
98.3079E-02	86.1382E-02

A 0 COEFFICIENTS

-78.1317E-01 -18.7741E-01

B 0 COEFFICIENTS

.0000E+00 .0000E+00

A 1 COEFFICIENTS

35.9970E-01 54.7646E-02

B 1 COEFFICIENTS

34.0609E-01 -14.3619E-01

A 2 COEFFICIENTS

27.2329E-02 67.4262E-02

B 2 COEFFICIENTS

38.9324E-02 -20.0546E-02

A 3 COEFFICIENTS

17.2081E-01 15.4650E-02

B 3 COEFFICIENTS

-10.6336E-01 15.0248E-02

A 4 COEFFICIENTS

-83.4205E-02 -14.0620E-02

B 4 COEFFICIENTS

-30.4409E-02 17.1890E-03

A 5 COEFFICIENTS

10.4539E-02 -28.8765E-03

B 5 COEFFICIENTS

-78.2825E-03 89.7057E-04

A 6 COEFFICIENTS

-22.6880E-02 -25.1819E-03

B 6 COEFFICIENTS

70.2810E-03 56.6429E-04

A 7 COEFFICIENTS

84.7070E-03 27.9216E-04

B 7 COEFFICIENTS

39.0608E-03 16.2973E-03

A 8 COEFFICIENTS

-38.8016E-03 -33.7700E-04

B 8 COEFFICIENTS

-14.5142E-03 35.6968E-04

A 9 COEFFICIENTS

15.6598E-03 86.7653E-05

B 9 COEFFICIENTS

25.5472E-03 89.6789E-04

A10 COEFFICIENTS

52.2396E-04	92.4518E-05

B10 COEFFICIENTS

-17.6781E-03	20.3526E-04

A11 COEFFICIENTS

-25.9063E-03	-10.4201E-04

B11 COEFFICIENTS

23.2306E-03	81.9764E-04

A12 COEFFICIENTS

77.0499E-03	85.9993E-04

B12 COEFFICIENTS

16.7042E-03	24.1281E-04

A13 COEFFICIENTS

-36.6687E-03	91.0647E-04

B13 COEFFICIENTS

-22.0450E-03	41.8428E-04

A14 COEFFICIENTS

28.0792E-02	46.8645E-03

B14 COEFFICIENTS

-10.3866E-02	61.0813E-04

A15 COEFFICIENTS

-57.5361E-02	-52.1643E-03

B15 COEFFICIENTS

-35.3607E-02	50.5087E-03

A16 COEFFICIENTS

-88.9147E-03	-22.5051E-02

B16 COEFFICIENTS

12.9291E-02	-66.6425E-03

A17 COEFFICIENTS

-12.0094E-01	-18.3170E-02

B17 COEFFICIENTS

11.3537E-01	-47.8633E-02

A18 COEFFICIENTS

26.0573E-01	62.5389E-02

B18 COEFFICIENTS

-47.2771E-10	-13.1795E-10

INCREMENT OF RUN EQUALS 360 STEPS

FOR THIS RUN:
B= .97 MU= 2.40 GAMMA=12.00 PSQUARE= 1.00

POINCARE EXPONENTS

OMEGA1=-29.86568E-01 OMEGA2= 56.85900E-03

SYSTEM OF EIGENVECTORS

-15.1443E-02	57.8038E-02
98.8466E-02	81.6010E-02

A 0 COEFFICIENTS

-11.0410E+00 -29.2664E-01

B 0 COEFFICIENTS

.0000E+00 .0000E+00

A 1 COEFFICIENTS

53.6173E-01 46.0498E-02

B 1 COEFFICIENTS

44.5590E-01 -30.6984E-02

A 2 COEFFICIENTS

-20.3946E-02 20.7150E-01

B 2 COEFFICIENTS

55.2562E-02 79.6890E-02

A 3 COEFFICIENTS

26.8111E-01 18.6141E-01

B 3 COEFFICIENTS

-17.2685E-01 76.1854E-03

A 4 COEFFICIENTS

-14.8115E-01	10.3161E-01

B 4 COEFFICIENTS

-32.3913E-02	-83.7459E-02

A 5 COEFFICIENTS

24.2456E-02	54.4349E-02

B 5 COEFFICIENTS

-15.3655E-02	-95.0396E-02

A 6 COEFFICIENTS

-39.6081E-02	20.8239E-02

B 6 COEFFICIENTS

18.1990E-02	-76.2580E-02

A 7 COEFFICIENTS

17.4706E-02	12.3798E-02

B 7 COEFFICIENTS

46.6590E-03	-55.9142E-02

A 8 COEFFICIENTS

-71.1219E-03	90.6268E-03

B 8 COEFFICIENTS

-95.5110E-04	-46.1019E-02

A 9 COEFFICIENTS

33.6707E-03	87.6234E-03

B 9 COEFFICIENTS

35.4174E-03	-39.6964E-02

A10 COEFFICIENTS

61.3447E-04	78.7251E-03

B10 COEFFICIENTS

-22.3023E-03	-37.9507E-02

A11 COEFFICIENTS

-50.9039E-03	60.1988E-03

B11 COEFFICIENTS

33.4580E-03	-37.3592E-02

A12 COEFFICIENTS

13.2496E-02	25.0669E-03

B12 COEFFICIENTS

49.5796E-03	-41.2000E-02

A13 COEFFICIENTS

-82.8849E-03	-94.0776E-03

B13 COEFFICIENTS

-42.8968E-03	-44.4325E-02

A14 COEFFICIENTS

49.7726E-02	-26.0973E-02

B14 COEFFICIENTS

-11.2193E-02	-38.0065E-02

A15 COEFFICIENTS

-89.6710E-02	-54.2180E-02

B15 COEFFICIENTS

-57.3149E-02	-49.1236E-03

A16 COEFFICIENTS

71.1015E-03	-61.4583E-02

B16 COEFFICIENTS

18.3251E-02	21.6284E-02

A17 COEFFICIENTS

-17.8941E-01	-79.6183E-03

B17 COEFFICIENTS

14.8565E-01	-12.6878E-02

A18 COEFFICIENTS

36.8272E-01	10.4914E-01

B18 COEFFICIENTS

-79.5457E-10	-11.3262E-10

INCREMENT OF RUN EQUALS 360 STEPS

FOR THIS RUN:
B= .97 MU= 2.40 GAMMA=14.00 PSQUARE= 1.00

POINCARE EXPONENTS

OMEGA1=-30.31760E-01 OMEGA2= 25.30000E-03

SYSTEM OF EIGENVECTORS

-11.3653E-02	64.3938E-02
99.3521E-02	76.5077E-02

A 0 COEFFICIENTS

-85.9898E+00 -19.1410E+00

B 0 COEFFICIENTS

.0000E+00 .0000E+00

A 1 COEFFICIENTS

26.8125E+00 34.9980E-01

B 1 COEFFICIENTS

48.4851E+00 27.8354E+00

A 2 COEFFICIENTS

12.5882E+00 37.0394E+00

B 2 COEFFICIENTS

68.6565E-01 23.7821E+00

A 3 COEFFICIENTS

24.7123E+00 47.2115E+00

B 3 COEFFICIENTS

-16.3433E+00 -34.4419E-01

A 4 COEFFICIENTS

-11.0840E+00	34.4015E+00

B 4 COEFFICIENTS

-97.9130E-01	-24.2313E+00

A 5 COEFFICIENTS

-70.0527E-02	17.4554E+00

B 5 COEFFICIENTS

-22.7835E-01	-28.1987E+00

A 6 COEFFICIENTS

-41.3841E-01	70.2036E-01

B 6 COEFFICIENTS

98.6141E-02	-23.2191E+00

A 7 COEFFICIENTS

12.0444E-01	32.4451E-01

B 7 COEFFICIENTS

50.0426E-02	-17.4411E+00

A 8 COEFFICIENTS

-48.3421E-02	24.6653E-01

B 8 COEFFICIENTS

-40.7154E-02	-13.8133E+00

A 9 COEFFICIENTS

29.3130E-02	24.1659E-01

B 9 COEFFICIENTS

-71.8964E-03	-11.9622E+00

A10 COEFFICIENTS

31.1714E-03	23.2357E-01

B10 COEFFICIENTS

-51.3457E-02	-11.2468E+00

A11 COEFFICIENTS

-34.7540E-02	18.9489E-01

B11 COEFFICIENTS

11.5931E-02	-11.4030E+00

A12 COEFFICIENTS

13.7792E-01	44.3689E-02

B12 COEFFICIENTS

89.1242E-03	-12.4071E+00

A13 COEFFICIENTS

21.3890E-02	-32.1465E-01

B13 COEFFICIENTS

-82.2281E-02	-13.2095E+00

A14 COEFFICIENTS

37.1995E-01	-90.1112E-01

B14 COEFFICIENTS

-33.9488E-01	-11.0791E+00

A15 COEFFICIENTS

-82.6538E-01	-13.3986E+00

B15 COEFFICIENTS

-55.0265E-01	-33.7420E-01

A16 COEFFICIENTS

-41.7859E-01	-10.0905E+00

B16 COEFFICIENTS

22.3279E-01	64.5452E-01

A17 COEFFICIENTS

-89.6017E-01	10.3887E-01

B17 COEFFICIENTS

16.1402E+00	85.4520E-01

A18 COEFFICIENTS

28.6747E+00	85.6943E-01

B18 COEFFICIENTS

-67.6235E-09	38.6049E-10

INCREMENT OF RUN EQUALS 360 STEPS

FOR THIS RUN;
B= .97 MU= 2.40 GAMMA=16.00 PSQUARE= 1.00

POINCARE EXPONENTS

OMEGA1=-33.36009E-01 OMEGA2=-10.56400E-03

SYSTEM OF EIGENVECTORS

-70.7829E-03	70.5935E-02
99.7492E-02	70.8276E-02

A 0 COEFFICIENTS

-23.8665E+01 -13.7204E+01

B 0 COEFFICIENTS

.0000E+00 .0000E+00

A 1 COEFFICIENTS

64.4310E+00 73.7865E+00

B 1 COEFFICIENTS

14.1890E+01 31.0854E+01

A 2 COEFFICIENTS

42.8193E+00 43.5389E+01

B 2 COEFFICIENTS

25.1191E+00 26.0912E+01

A 3 COEFFICIENTS

75.7699E+00 56.9025E+01

B 3 COEFFICIENTS

-49.0324E+00 -45.6522E+00

A 4 COEFFICIENTS

-32.8602E+00	43.5564E+01

B 4 COEFFICIENTS

-33.1830E+00	-29.5376E+01

A 5 COEFFICIENTS

-44.0871E-01	22.9534E+01

B 5 COEFFICIENTS

-76.0662E-01	-35.7071E+01

A 6 COEFFICIENTS

-13.9459E+00	92.3757E+00

B 6 COEFFICIENTS

41.4565E-01	-30.1447E+01

A 7 COEFFICIENTS

38.2142E-01	38.4888E+00

B 7 COEFFICIENTS

22.6164E-01	-22.6983E+01

A 8 COEFFICIENTS

-13.4716E-01	27.8865E+00

B 8 COEFFICIENTS

-75.0055E-02	-17.6813E+01

A 9 COEFFICIENTS

10.5338E-01	29.1415E+00

B 9 COEFFICIENTS

-24.2361E-02	-15.1420E+01

A10 COEFFICIENTS

84.3313E-04	29.9560E+00

B10 COEFFICIENTS

-13.3666E-01	-14.2526E+01

A11 COEFFICIENTS

-10.7501E-01	25.0097E+00

B11 COEFFICIENTS

60.8091E-02	-14.6181E+01

A12 COEFFICIENTS

45.9554E-01	48.6618E-01

B12 COEFFICIENTS

68.7915E-02	-15.9560E+01

A13 COEFFICIENTS

14.1300E-01	-43.0647E+00

B13 COEFFICIENTS

-26.9712E-01	-16.7415E+01

A14 COEFFICIENTS

11.0020E+00	-11.3643E+01

B14 COEFFICIENTS

-11.4450E+00	-13.6665E+01

A15 COEFFICIENTS

-25.3529E+00	-15.9643E+01

B15 COEFFICIENTS

-16.4896E+00	-43.5968E+00

A16 COEFFICIENTS

-14.2387E+00	-11.6184E+01

B16 COEFFICIENTS

82.0835E-01	68.1757E+00

A17 COEFFICIENTS

-21.5640E+00	36.9317E-01

B17 COEFFICIENTS

47.2386E+00	94.2555E+00

A18 COEFFICIENTS

79.5737E+00	73.8045E+00

B18 COEFFICIENTS

-20.4307E-08	58.6715E-09

APPENDIX E

```
100 REM 4TH ORDER RUNGA-KUTTA ROUTINE FOR UP TO AN 8TH ORDER SYSTEM
110 REM
120 CALL CLEAR
130 REM
140 OPEN #1:"PIO"
150 REM
160 DIM X(8,8),XWORK(8,8),XNEW(8,8),AMAT(8,8)
170 DIM F1(8,8),F2(8,8),F3(8,8),F4(8,8),F(8,8)
180 DIM K1(8,8),K2(8,8),K3(8,8),K4(8,8)
190 REM
200 REM INITIALIZE ALL VARIABLES
210 REM
220 ANGLE=0.0
230 FOR I=1 TO 8
240 FOR J=1 TO 8
250 X(I,J)=0.0
260 IF I=J THEN X(I,J)=1.0
270 NEXT J
280 NEXT I
290 REM
300 CALL CLEAR
310 REM
320 REM ENTER THE  INPUT PARAMETERS
330 REM
340 INPUT "ENTER # OF INCREMENTS":INCR
350 DELTANGLE=2*PI/INCR
360 REM
370 INPUT "ENTER ORDER OF SYSTEM":ORDER
372 INPUT "ENTER IN B:":BE
374 INPUT "ENTER IN MU:":MU
376 REM INPUT "ENTER IN GAMMA:":GAMMA
378 INPUT "ENTER IN PSQUARE:":PSQUARE
379 CALL CLEAR
380 REM
390 REM MASTER LOOP
400 REM
405 FOR GAMMA=2 TO 16 STEP 2
406 DISPLAY AT(5,4):"GAMMA=",GAMMA
410 FOR N=1 TO INCR STEP 1
420 DISPLAY AT(2,4):"N=",N
430 REM
440 REM FIRST RUNGA-KUTTA CONSTANT
450 REM
460 CALL FCNN(ANGLE,X(,),F1(,),BE,MU,GAMMA,PSQUARE)
470 REM
480 FOR I=1 TO ORDER
490 FOR J=1 TO ORDER
500 K1(I,J)=DELTANGLE*F1(I,J)
530 REM
```

```
560 XWORK(I,J)=X(I,J)+0.5*K1(I,J)
570 NEXT J
580 NEXT I
590 REM
600 REM SECOND RUNGA-KUTTA CONSTANT
610 REM
620 CALL FCNN(ANGLE+0.5*DELTANGLE,XWORK(,),F2(,),BE,MU,GAMMA,PSQUARE)
630 REM
640 FOR I=1 TO ORDER
650 FOR J=1 TO ORDER
660 K2(I,J)=DELTANGLE*F2(I,J)
690 REM
720 XWORK(I,J)=X(I,J)+0.5*K2(I,J)
730 NEXT J
740 NEXT I
750 REM
760 REM THIRD RUNGA-KUTTA CONSTANT
770 REM
780 CALL FCNN(ANGLE+0.5*DELTANGLE,XWORK(,),F3(,),BE,MU,GAMMA,PSQUARE)
790 REM
800 FOR I=1 TO ORDER
810 FOR J=1 TO ORDER
820 K3(I,J)=DELTANGLE*F3(I,J)
850 REM
880 XWORK(I,J)=X(I,J)+K3(I,J)
890 NEXT J
900 NEXT I
910 REM
920 REM FOURTH RUNGA-KUTTA CONSTANT
930 REM
940 CALL FCNN(ANGLE+DELTANGLE,XWORK(,),F4(,),BE,MU,GAMMA,PSQUARE)
950 REM
960 FOR I=1 TO ORDER
970 FOR J=1 TO ORDER
980 K4(I,J)=DELTANGLE*F4(I,J)
990 NEXT J
1000 NEXT I
1010 REM
1020 REM EVALUATION OF STATE VARIABLES
1030 REM
1040 FOR I=1 TO ORDER
1050 FOR J=1 TO ORDER
1060 XNEW(I,J)=X(I,J)+(1/6)*(K1(I,J)+2*K2(I,J)+2*K3(I,J)+K4(I,J))
1070 NEXT J
1080 NEXT I
1090 REM
1100 REM UPDATE STATE VARIABLES
1110 REM
1120 FOR I=1 TO ORDER
1130 FOR J=1 TO ORDER
1140 X(I,J)=XNEW(I,J)
1150 NEXT J
1160 NEXT I
```

```
1360 SUB FCNN(T,XWORK(,),F(,),BE,MU,GAMMA,PSQUARE)
1370 REM
1380 CALL MATRIXA(T,AMAT(,),BE,MU,GAMMA,PSQUARE)
1390 REM
1400 CALL MATRIXMULT(AMAT(,),XWORK(,),2,2,2,F(,))
1410 REM
1420 REM
1430 SUBEND
2000 SUB MATRIXA(PSI,MATA(,),BE,MU,GAMMA,PSQUARE)
2001 REM
2002 REM ALGORITHM TO CALCULATE STATE TRANSITION MATRIX
2003 REM FOR A PERIODIC FUNCTION
2004 REM IN PARTICULAR-HELICOPTER BLADE, NO FEEDBACK
2005 REM  UPDATED 16 AUG 1984
2006 REM
2007 DEF ARCSIN(ZZ)=ATN(ZZ/SQR(1-ZZ*ZZ))
2008 PSIT=PSI
2010 IF PSIT>=0.0 AND PSIT<PI THEN 2020 ELSE 2012
2012 IF PSIT>=PI AND PSIT<(PI+ARCSIN(BE/MU))THEN 2030 ELSE 2014
2014 IF PSIT>=(2*PI-ARCSIN(BE/MU))AND PSIT<(2*PI)-.000001 THEN 2030 ELSE 2016
2016 IF PSIT>(2*PI)-.000001 THEN PSIT=0.0 :: GO TO 2020 :: ELSE GO TO 2018
2018 GO TO 2040
2020 REM PRINT #1:"2020"
2022 K=(1/3)*BE^3*MU*COS(PSIT)+(1/4)*BE^2*MU^2*SIN(2*PSIT)
2023 REM
2024 C=(1/4)*BE^4+(1/3)*BE^3*MU*SIN(PSIT)
2025 REM
2026 GO TO 2050
2030 REM PRINT #1:"2030"
2031 REM
2032 K=(1/3)*BE^3*MU*COS(PSIT)+(1/4)*BE^2*MU^2*SIN(2*PSIT)+MU^4*(-(1/12)*SIN(2*P
SIT)+(1/24)*SIN(4*PSIT))
2033 REM
2034 C=(1/4)*BE^4+(1/3)*BE^3*MU*SIN(PSIT)+MU^4*((1/16)-(1/12)*COS(2*PSIT)+(1/48)
*COS(4*PSIT))
2035 REM
2036 GO TO 2050
2040 REM PRINT #1:"2040"
2041 REM
2042 K=-(1/3)*BE^3*MU*COS(PSIT)-(1/4)*BE^2*MU^2*SIN(2*PSIT)
2043 REM
2044 C=-(1/4)*BE^4-(1/3)*BE^3*MU*SIN(PSIT)
2045 REM
2050 REM CONSTRUCT MATRIX A
2051 REM
2052 MATA(1,1)=0
2053 MATA(1,2)=1
2054 MATA(2,1)=-(GAMMA/2)*(2*PSQUARE/GAMMA+K)
2055 MATA(2,2)=-(GAMMA*C/2)
2056 REM
2057 REM RETURN TO MAIN PROGRAM
2060 SUBEND
```

```
1170 REM
1180 ANGLE=ANGLE+DELTANGLE
1190 REM
1200 NEXT N
1210 REM
1220 REM PRINT OUTPUT
1230 REM
1231 PRINT #1:"-----------------------------------------------------------
---"
1232 PRINT #1
1233 PRINT #1,USING "  BE=##.##^^^^   MU=##.##^^^^":BE,MU
1234 PRINT #1
1235 PRINT #1,USING "  GAMMA=##.##^^^^   PSQUARE=##.##^^^^":GAMMA,PSQUARE
1236 PRINT #1
1237 PRINT #1
1240 FOR I=1 TO ORDER
1250 FOR J=1 TO ORDER
1260 PRINT #1,USING "    X(#,#)=###.#####^^^^":I,J,X(I,J)
1270 NEXT J
1280 NEXT I
1290 PRINT #1
1300 PRINT #1
1310 PRINT #1
1320 REM
1325 NEXT GAMMA
1326 REM
1330 CLOSE #1
1340 REM
1350 END
```

```
7000 SUB MATRIXMULT(A(,),B(,),ROWA,COLA,COLB,OUT(,))
7001 REM
7002 REM MATRIX MULTIPLICATION ROUTINE
7003 REM INPUTS    MATRIX "A", MATRIX "B",DIMENSION OF AAND COLUMN DIMENSION OF
B
7004 FOR I=1 TO ROWA
7005 FOR J=1 TO COLA
7006 OUT(I,J)=0.0
7007 NEXT J
7008 NEXT I
7009 REM
7010 FOR I=1 TO ROWA
7020 FOR K=1 TO COLB
7030 FOR J=1 TO COLA
7040 SUM=A(I,J)*B(J,K)
7050 OUT(I,K)=OUT(I,K)+SUM
7060 NEXT J
7070 NEXT K
7080 NEXT I
7090 REM
7999 SUBEND :: REM RETURN TO MAIN PROGRAM
```

APPENDIX F

```
100 REM  PROGRAM EIGENP- EIGENVALUE/EIGENVECTOR ROUTINE
110 REM
120 OPEN #1:"PIO"
125 CALL CLEAR
130 DIM A(4,4),VECR(4,4),VECI(4,4),EVR(4),EVI(4),INDIC(4)
140 DIM IWORK(10),LOCAL(10),PRFACT(10),SUBDIA(10)
150 DIM WORK1(10),WORK2(10),WORK(10),H(4,4)
170 REM
180 PRINT "ENTER IN A MATRIX"
190 CALL INPUTT(N,NM,A(,),IERROR)
200 REM
210 T=64
220 REM
230 FOR I=1 TO N
235 FOR J=1 TO N
240 PRINT #1,USING "   A(##,##)=##.######^^^^":I,J,A(I,J)
245 NEXT J
250 NEXT I
255 REM
256 PRINT #1
257 PRINT #1
258 PRINT #1
260 REM
360 REM
370 IF N<>1 THEN 450
380 EVR(1)=A(1,1)
390 EVI(1)=0.0
400 VECR(1,1)=1.0
410 VECI(1,1)=0.0
420 INDIC(1)=2
430 GO TO 2080
440 REM  STATEMENT LABEL 1
450 CALL SCALEE(N,NM,A(,),VECI(,),PRFACT(),ENORM)
460 REM THE COMPUTATION OF THE EIGENVALUES OF A NORMALIZED MATRIX
470 REM
480 EX=EXP(-T*LOG(2.0))
490 CALL HESQR(N,NM,A(,),VECI(,),EVR(),EVI(),SUBDIA(),INDIC(),EPS,EX)
500 J=N
510 I=1
520 LOCAL(1)=1
530 IF J=1 THEN 650
540 REM STATEMENT LABEL 2
550 IF ABS(SUBDIA(J-1))>EPS THEN 580
560 I=I+1
570 LOCAL(I)=0.
580 REM STATEMENT LABEL 3
590 J=J-1
600 LOCAL(I)=LOCAL(I)+1
610 IF J<>1 THEN 540
```

```
620 REM
630 REM EIGENVECTOR PROBLEM
640 REM
650 REM STATEMENT LABEL 4
660 K=1
670 KON=0.
680 L=LOCAL(1)
690 M=N
700 FOR I=1 TO N
710 IVEC=N-I+1
720 IF I<=L THEN 760
730 K=K+1
740 M=N-L
750 L=L+LOCAL(K)
760 REM STATEMENT LABEL 5
770 IF INDIC(IVEC)=0 THEN 1070
780 IF EVI(IVEC)<>0.0 THEN 1000
790 REM
800 REM TRANSFER OF AN UPPER-HESSENBERG MATRIX OF THE ORDER
810 REM M FROM THE ARRAYS VECI AND SUBDIA INTO ARRAY A
820 REM
830 FOR K1=1 TO M
840 FOR L1=K1 TO M
850 A(K1,L1)=VECI(K1,L1)
860 NEXT L1
870 IF K1=1 THEN 890
880 A(K1,K1-1)=SUBDIA(K1-1)
890 REM STATEMENT LABEL 7
900 NEXT K1
910 REM
920 REM THE COMPUTATION OF THE REAL EIGENVECTOR IVEC FOR THE UPPER HESSENBERG MA
TRIX
930 REM CORRESPONDING TO THE REAL EIGENVALUE EVR(IVEC)
940 REM
950 CALL REALVE(N,NM,M,IVEC,A(,),VECR(,),EVR(),EVI(),IWORK(),WORK(),INDIC(),EPS,
EX)
960 GO TO 1070
970 REM
980 REM THE COMPUTATION OF THE COMPLEX EIGENVECTOR
990 REM
1000 REM STATEMENT LABEL 8
1010 IF KON<>0 THEN 1050
1020 KON=1
1030 CALL COMPVE(N,NM,M,IVEC,A(,),VECR(,),VECI(,),EVR(),EVI(),INDIC(),IWORK(),SU
BDIA(),WORK1(),WORK2(),WORK(),EPS,EX)
1040 GO TO 1070
1050 REM STATEMENT LABEL 9
1060 KON=0
1070 REM STATEMENT LABEL 10
1075 NEXT I
1080 REM
1090 REM
1100 REM THE RECONSTRUCTION OF THE MATRIX
1110 REM
```

```
1120 FOR I=1 TO N
1130 FOR J=1 TO N
1140 A(I,J)=0.0
1150 A(J,I)=0.0
1160 NEXT J
1180 A(I,I)=1.
1190 NEXT I
1200 IF N<=2 THEN 1370
1210 M=N-2
1220 FOR K=1 TO M
1230 L=K+1
1240 FOR J=2 TO N
1250 D1=0.0
1260 FOR I=L TO N
1270 D2=VECI(I,K)
1280 D1=D1+D2*A(J,I)
1290 NEXT I
1300 FOR I=L TO N
1310 A(J,I)=A(J,I)-VECI(I,K)*D1
1320 NEXT I
1325 NEXT J
1330 NEXT K
1340 REM
1350 REM COMPUTE THE EIGENVECTORS OF THE ORIGINAL NON-SCALED MATRIX
1360 REM
1370 REM STATEMENT LABEL 15
1380 KON=1
1390 FOR I=1 TO N
1400 L=0.
1410 IF EVI(I)=0.0 THEN 1460
1420 L=1
1430 IF KON=0.0 THEN 1460
1440 KON=0
1450 GO TO 2010
1460 REM STATEMENT LABEL 16
1470 FOR J=1 TO N
1480 D1=0.0
1490 D2=0.0
1500 FOR K=1 TO N
1510 D3=A(J,K)
1520 D1=D1+D3*VECR(K,I)
1530 IF L=0 THEN 1550
1540 D2=D2+D3*VECR(K,I-1)
1550 NEXT K :: REM STATEMENT LABEL 17
1560 WORK(J)=D1/PRFACT(J)
1570 IF L=0.0 THEN 1590
1580 SUBDIA(J)=D2/PRFACT(J)
1590 REM STATEMENT LABEL 18
1600 NEXT J
1610 REM
1620 REM THE NORMALIZATION OF THE EIGENVECTORS AND THE COMPUTATION OF THE EIGEN
ALUES
1630 REM OF THE ORIGINAL NON-NORMALIZED MATRIX.
```

```
1640 IF L=1 THEN 1760
1650 D1=0.0
1660 FOR M=1 TO N
1670 D1=D1+WORK(M)^2
1680 NEXT M
1690 D1=SQR(D1)
1700 FOR M=1 TO N
1710 VECI(M,I)=0.0
1720 VECR(M,I)=WORK(M)/D1
1730 NEXT M
1740 EVR(I)=EVR(I)*ENORM
1750 GO TO 2010
1760 REM SRSTATEMENT LABEL 21
1770 KON=1
1780 EVR(I)=EVR(I)*ENORM
1790 EVR(I-1)=EVR(I)
1800 EVI(I)=EVI(I)*ENORM
1810 EVI(I-1)=-EVI(I)
1820 R=0.0
1830 FOR J=1 TO N
1840 R1=WORK(J)^2+SUBDIA(J)^2
1850 IF R>=R1 THEN 1880
1860 R=R1
1870 L=J
1880 REM STATEMENT LABEL 22
1890 NEXT J
1900 D3=WORK(L)
1910 R1=SUBDIA(L)
1920 FOR J=1 TO N
1930 D1=WORK(J)
1940 D2=SUBDIA(J)
1950 VECR(J,I)=(D1*D3+D2*R1)/R
1960 VECI(J,I)=(D2*D3-D1*R1)/R
1970 VECR(J,I-1)=VECR(J,I)
1980 REM STATEMENT 23
1990 VECI(J,I-1)=-VECI(J,I)
2000 NEXT J
2010 REM STATEMENT 24
2020 NEXT I
2070 REM
2080 REM STATEMENT 25
2081 FOR I=1 TO N
2082 PRINT #1,USING "EIGENVALUE ## EQUALS ##.####^^^^ + ##.####^^^^J":I,EVR(I),E
VI(I)
2083 PRINT #1,USING "EIGENVECTOR EQUALS"
2084 FOR J=1 TO N
2085 PRINT #1,USING " ###.#####^^^^ + ###.#####^^^^J":VECR(J,I),VECI(J,I)
2086 NEXT J
2087 PRINT #1
2088 NEXT I
2089 CLOSE #1
2090 END
```

```
3000 SUB SCALEE(N,NM,A(,),H(,),PRFACT(),ENORM)
3010 REM
3020 REM
3030 FOR II=1 TO N
3040 FOR JJ=1 TO N
3050 H(II,JJ)=A(II,JJ)
3060 NEXT JJ
3070 PRFACT(II)=1.0
3080 NEXT II
3090 BOUND1=0.75
3100 BOUND2=1.33
3110 ITER=0
3120 REM STATEMENT LABEL 3
3130 NCOUNT=0.
3140 FOR II=1 TO N
3150 COLUMN=0.0
3160 ROW=0.0
3170 FOR JJ=1 TO N
3180 IF II=JJ THEN 3210
3190 COLUMN=COLUMN+ABS(A(JJ,II))
3200 ROW=ROW+ABS(A(II,JJ))
3210 REM STATEMENT 4
3220 NEXT JJ
3230 IF COLUMN=0.0 THEN 3280
3240 IF ROW=0.0 THEN 3280
3250 Q=COLUMN/ROW
3260 IF Q<BOUND1 THEN 3310
3270 IF Q>BOUND2 THEN 3310
3280 REM STATEMENT LABEL 5
3290 NCOUNT=NCOUNT+1
3300 GO TO 3400
3310 REM STATEMENT LABEL 6
3320 FACTOR=SQR(Q)
3330 FOR JJ=1 TO N
3340 IF II=JJ THEN 3370
3350 A(II,JJ)=A(II,JJ)*FACTOR
3360 A(JJ,II)=A(JJ,II)/FACTOR
3370 REM STATEMENT LABEL 7
3380 NEXT JJ
3390 PRFACT(II)=PRFACT(II)*FACTOR
3400 REM STATEMENT LABEL 8
3410 NEXT II
3420 ITER=ITER+1
3430 IF ITER>30 THEN 3610
3440 IF NCOUNT<N THEN 3120
3450 FNORM=0.0
3460 FOR II=1 TO N
3470 FOR JJ=1 TO N
3480 Q=A(II,JJ)
3490 FNORM=FNORM+Q*Q
3500 REM STATEMENT LABEL 9
3510 NEXT JJ
3520 NEXT II
```

```
3525 FNORM=SQR(FNORM)
3530 FOR II=1 TO N
3540 FOR JJ=1 TO N
3550 A(II,JJ)=A(II,JJ)/FNORM
3560 REM STATEMNET LABEL 10
3570 NEXT JJ
3580 NEXT II
3590 ENORM=FNORM
3600 GO TO 3690
3610 REM STATEMENT LABEL 11
3620 FOR II=1 TO N
3625 PRFACT(II)=1.0
3630 FOR JJ=1 TO N
3640 A(II,JJ)=H(II,JJ)
3650 REM STATEMENT LABEL 12
3660 NEXT JJ
3670 NEXT II
3680 ENORM=1.0
3690 REM STATEMENT LABEL 13
3700 REM
3710 SUBEND
```

```
4000 SUB HESQR(N,NM,A(,),H(,),EVR(),EVI(),SUBDIA(),INDIC(),EPS,EX)
4010 REM
4020 REM
4030 IF N-2<0 THEN 4780
4040 IF N-2=0 THEN 4060
4050 IF N-2>0 THEN 4090
4060 REM STATEMENT LABEL 1
4070 SUBDIA(1)=A(2,1)
4080 GO TO 4780 :: REM GO TO 14
4090 REM STATEMENT LABEL 2
4100 M=N-2
4110 FOR KK=1 TO M
4120 L=KK+1
4130 S=0.0
4140 FOR III=L TO N
4150 H(III,KK)=A(III,KK)
4160 REM STATEMENT LABEL 3
4170 S=S+ABS(A(III,KK))
4175 NEXT III
4180 IF S<>ABS(A(KK+1,KK))THEN 4220 :: REM GO TO 4
4190 SUBDIA(KK)=A(KK+1,KK)
4200 H(KK+1,KK)=0.0
4210 GO TO 4690 :: REM GO TO 12
4220 REM STATEMENT LABEL 4
4230 SR2=0.0
4240 FOR III=L TO N
4250 SR=A(III,KK)
4260 SR=SR/S
4270 A(III,KK)=SR
4280 REM STATEMENT LABEL 5
4290 SR2=SR2+SR*SR
4300 NEXT III
4310 SR=SQR(SR2)
4320 IF A(L,KK)<0.0 THEN 4340 :: REM GO TO 6
4330 SR=-SR
4340 REM STATEMENT LABEL 6
4350 SR2=SR2-SR*A(L,KK)
4360 A(L,KK)=A(L,KK)-SR
4370 H(L,KK)=H(L,KK)-SR*S
4380 SUBDIA(KK)=SR*S
4390 X=S*SQR(SR2)
4400 FOR III=L TO N
4410 H(III,KK)=H(III,KK)/X
4420 REM STATEMENT LABEL 7
4430 SUBDIA(III)=A(III,KK)/SR2
4440 NEXT III
4450 REM
4460 REM PREMULTIPLICATION BY THE MATRIX PR.
4470 REM
4480 FOR JJJ=L TO N
4490 SR=0.0
```

```
4500 FOR III=L TO N
4510 SR=SR+A(III,KK)*A(III,JJJ)
4520 NEXT III
4530 FOR III=L TO N
4540 A(III,JJJ)=A(III,JJJ)-SUBDIA(III)*SR
4550 NEXT III
4560 NEXT JJJ
4570 REM
4580 REM POSTMULTIPLICATION BY THE MATRIX PR.
4590 REM
4600 FOR JJJ=1 TO N
4610 SR=0.0
4620 FOR III=L TO N
4630 SR=SR+A(JJJ,III)*A(III,KK)
4640 NEXT III
4650 FOR III=L TO N
4660 A(JJJ,III)=A(JJJ,III)-SUBDIA(III)*SR
4670 NEXT III
4680 NEXT JJJ
4690 REM STATEMENT LABEL 12
4700 NEXT KK
4710 FOR KK=1 TO M
4720 A(KK+1,KK)=SUBDIA(KK)
4730 NEXT KK
4740 REM TRANSFER OF THE UPPER HALF OF THE MATRIX A INTO THE ARRAY H
4750 REM AND THE CALCULATION OF THE SMALL POSITIVE NUMBER EPS.
4760 REM
4770 SUBDIA(N-1)=A(N,N-1)
4780 REM STATEMENT LABEL 14
4790 EPS=0.0
4800 FOR KK=1 TO N
4810 INDIC(KK)=0
4820 IF KK<>N THEN EPS=EPS+SUBDIA(KK)^2
4830 FOR III=KK TO N
4840 H(KK,III)=A(KK,III)
4850 REM STATEMENT LABEL 15
4855 EPS=EPS+A(KK,III)^2
4860 NEXT III
4870 NEXT KK
4880 EPS=EX*SQR(EPS)
4890 REM
4900 REM THE QR ITERATIVE PROCESS
4910 REM
4920 SHIFT=A(N,N-1)
4930 IF N<=2 THEN SHIFT=0.0
4940 IF A(N,N)<>0.0 THEN SHIFT=0.0
4950 IF A(N-1,N)<>0.0 THEN SHIFT=0.0
4960 IF A(N-1,N-1)<>0.0 THEN SHIFT=0.0
4970 M=N
4980 NS=0
4990 MAXST=N*10
5000 REM
```

```
5010 REM TESTING IF THE UPPER HALF OF THE MATRIX IS EQUAL TO ZERO.
5020 REM IF IT IS EQUAL TO ZERO THE QR PROCESS IS NOT NECESSARY.
5030 REM
5040 FOR III=2 TO N
5050 FOR KK=III TO N
5060 IF A(III-1,KK)<>0.0 THEN 5180 :: REM GO TO 18
5070 NEXT KK
5080 NEXT III
5090 FOR III=1 TO N
5100 INDIC(III)=1
5110 EVR(III)=A(III,III)
5120 EVI(III)=0.0
5130 NEXT III
5140 GO TO 6620 :: REM GO TO 37
5150 REM
5160 REM START THE MAIN LOOP OF THE QR PROCESS
5170 REM
5180 REM STATEMENT LABEL 18
5190 K=M-1
5200 M1=K
5210 I=K
5220 REM FIND THE DECOMPOSITION OF THE MATRIX
5230 IF K<0 THEN 6620 :: REM STATEMENT 37
5240 IF K=0 THEN 6300 :: REM STATEMENT 34
5250 IF K>0 THEN 5260 :: REM STATEMENT 19
5260 REM STATEMENT LABEL 19
5270 IF ABS(A(M,K))<=EPS THEN 6300 :: REM GO TO STATEMENT 34
5280 IF M-2=0. THEN 6390 :: REM GO TO 35
5290 REM STATEMENT LABEL 20
5300 I=I-1
5310 IF ABS(A(K,I))<=EPS THEN 5340 :: REM GO TO 21
5320 K=I
5330 IF K>1 THEN 5290 :: REM GO TO 20
5340 REM STATEMENT LABEL 21
5350 IF K=M1 THEN 6390 :: REM GO TO 35
5360 REM TRANSFORMATION OF A MATRIX OF THE ORDER GREATER THAN 2
5370 REM
5380 S=A(M,M)+A(M1,M1)+SHIFT
5390 SR=A(M,M)*A(M1,M1)-A(M,M1)*A(M1,M)+0.25*SHIFT^2
5400 A(K+2,K)=0.0
5410 REM
5420 REM CALCULATE X1,Y1,Z1 FOR THE SUBMATRIX OBTAINED BY THE DECOMPOSITION
5430 REM
5440 X=A(K,K)*(A(K,K)-S)+A(K,K+1)*A(K+1,K)+SR
5450 Y=A(K+1,K)*(A(K,K)+A(K+1,K+1)-S)
5460 R=ABS(X)+ABS(Y)
5470 IF R=0.0 THEN SHIFT=A(M,M-1)
5480 IF R=0.0 THEN 5340 :: REM GO TO 21
5490 Z=A(K+2,K+1)*A(K+1,K)
5500 SHIFT=0.0
5510 NS=NS+1
5520 REM
```

```
5530 REM THE LOOP FOR ONE STEP OF THE QR PROCESS.
5540 REM
5550 FOR I=K TO M1
5560 IF I=K THEN 5630 :: REM GO TO 22
5570 REM CALCULATE XR,YR,ZR.
5580 X=A(I,I-1)
5590 Y=A(I+1,I-1)
5600 Z=0.0
5610 IF I+2>M THEN 5630 :: REM GO TO 22
5620 Z=A(I+2,I-1)
5630 REM STATEMENT 22
5640 SR2=ABS(X)+ABS(Y)+ABS(Z)
5650 IF SR2=0.0 THEN 5690 :: REM GO TO 23
5660 X=X/SR2
5670 Y=Y/SR2
5680 Z=Z/SR2
5690 REM STATEMENT LABEL 23
5700 S=SQR(X*X+Y*Y+Z*Z)
5710 IF X<0.0 THEN 5730 :: REM GO TO 24
5720 S=-S
5730 REM STATEMENT LABEL 24
5740 IF I=K THEN 5760 :: REM GO TO 25
5750 A(I,I-1)=S*SR2
5760 REM STATEMENT LABEL 25
5770 IF SR2<>0.0 THEN 5800 :: REM GO TO 26
5780 IF I+3>M THEN 6230 :: REM GO TO 33
5790 GO TO 6190 :: REM GO TO 32
5800 REM STATEMENT LABEL 26
5810 SR=1.0-X/S
5820 S=X-S
5830 X=Y/S
5840 Y=Z/S
5850 REM PREMULTIPLY BY THE MATRIX PR.
5860 REM
5870 FOR J=I TO M
5880 S=A(I,J)+A(I+1,J)*X
5890 IF I+2>M THEN 5910 :: REM GO TO 27
5900 S=S+A(I+2,J)*Y
5910 REM STATEMENT LABEL 27
5920 S=S*SR
5930 A(I,J)=A(I,J)-S
5940 A(I+1,J)=A(I+1,J)-S*X
5950 IF I+2>M THEN 5970 :: REM GO TO 28
5960 A(I+2,J)=A(I+2,J)-S*Y
5970 REM STATEMENT 28
5980 NEXT J
5990 REM POSTMULTIPLY BY THE MATRIX PR.
6000 REM
6010 L=I+2
6020 IF I<M1 THEN 6040 :: REM GO TO 29
6030 L=M
6040 REM STATEMENT LABEL 29
```

```
6050 FOR J=K TO L
6060 S=A(J,I)+A(J,I+1)*X
6070 IF I+2>M THEN 6090 :: REM GO TO 30
6080 S=S+A(J,I+2)*Y
6090 REM STATEMENT LABEL 30
6100 S=S*SR
6110 A(J,I)=A(J,I)-S
6120 A(J,I+1)=A(J,I+1)-S*X
6130 IF I+2>M THEN 6150 :: REM GO TO 31
6140 A(J,I+2)=A(J,I+2)-S*Y
6150 REM STATEMENT 31
6160 NEXT J
6170 IF I+3>M THEN 6230 :: REM GO TO 33
6180 S=-A(I+3,I+2)*Y*SR
6190 REM STATEMENT 32
6200 A(I+3,I)=S
6210 A(I+3,I+1)=S*X
6220 A(I+3,I+2)=S*Y+A(I+3,I+2)
6230 REM STATEMENT 33
6240 NEXT I
6250 REM
6260 IF NS>MAXST THEN 6620 :: REM GO TO 37
6270 GO TO 5180 :: REM GO TO 18
6280 REM
6290 REM COMPUTE THE LAST EIGENVALUE
6300 REM STATEMENT 34
6310 EVR(M)=A(M,M)
6320 EVI(M)=0.0
6330 INDIC(M)=1
6340 M=K
6350 GO TO 5180 :: REM GO TO 18
6360 REM
6370 REM COMPUTE THE EIGENVALUES OF THE LAST 2X2 MATRIX OBTAINED BY THE DECOMPOS
ITION
6380 REM
6390 REM STATEMENT LABEL 35
6400 R=0.5*(A(K,K)+A(M,M))
6410 S=0.5*(A(M,M)-A(K,K))
6420 S=S*S+A(K,M)*A(M,K)
6430 INDIC(K)=1
6440 INDIC(M)=1
6450 IF S<0.0 THEN 6530 :: REM GO TO 36
6460 T=SQR(S)
6470 EVR(K)=R-T
6480 EVR(M)=R+T
6490 EVI(K)=0.0
6500 EVI(K)=0.0
6510 M=M-2
6520 GO TO 5180 :: REM GO TO 18
6530 REM STATEMENT 36
```

```
6540 T=SQR(-S)
6550 EVR(K)=R
6560 EVI(K)=T
6570 EVR(M)=R
6580 EVI(M)=-T
6590 M=M-2
6600 GO TO 5180 :: GO TO 18
6610 REM
6620 REM STATEMENT 37
6630 SUBEND
```

```
7000 SUB REALVE(N,NM,M,IVEC,A(,),VECR(,),EVR(),EVI(),IWORK(),WORK(),INDIC(),EPS,
EX)
7010 REM
7020 REM
7030 VECR(1,IVEC)=1.0
7040 IF M=1 THEN 8340 :: REM GO TO 24
7050 EVALUE=EVR(IVEC)
7060 IF IVEC=M THEN 7160 :: REM GO TO 2
7070 K=IVEC+1
7080 R=0.0
7090 FOR II=K TO M
7100 IF EVALUE<>EVR(II)THEN 7130 :: REM GO TO 1
7110 IF EVI(II)<>0.0 THEN 7130 :: REM GO TO 1
7120 R=R+3.0
7130 REM STATEMENT 1
7140 NEXT II
7150 EVALUE=EVALUE+R*EX
7160 REM STATEMENT 2
7170 FOR K=1 TO M
7180 A(K,K)=A(K,K)-EVALUE
7190 NEXT K
7200 REM
7210 REM GAUSSIAN ELIMINATION OF UPPER-HESSENBERG MATRIX A
7220 REM
7230 K=M-1
7240 FOR II=1 TO K
7250 L=II+1
7260 IWORK(II)=0
7270 IF A(II+1,II)<>0.0 THEN 7310 :: REM GO TO 4
7280 IF A(II,II)<>0.0 THEN 7460 :: REM GO TO 8
7290 A(II,II)=EPS
7300 GO TO 7460 :: REM GO TO 8
7310 REM STATEMENT 4
7320 IF ABS(A(II,II))>=ABS(A(II+1,II))THEN 7400 :: REM GO TO 6
7330 IWORK(II)=1
7340 FOR J=II TO M
7350 R=A(II,J)
7360 A(II,J)=A(II+1,J)
7370 REM STATEMENT 5
7380 A(II+1,J)=R
7390 NEXT J
7400 REM STATEMENT 6
7410 R=-A(II+1,II)/A(II,II)
7420 A(II+1,II)=R
7430 FOR J=L TO M
7440 A(II+1,J)=A(II+1,J)+R*A(II,J)
7450 NEXT J
7460 REM STATEMENT 8
7465 NEXT II
7470 REM
7480 IF A(M,M)<>0.0 THEN 7520 :: REM GO TO 9
7490 A(M,M)=EPS
7500 REM
```

```
7510 REM THE VECTOR (1,1,...) IS STORED IN THE PLACE OF THE RIGHT HAND SIDE COLU
MN VECTOR
7520 REM STATEMENT 9
7530 FOR II=1 TO N
7540 IF II>M THEN 7570 :: REM GO TO 10
7550 WORK(II)=1.0
7560 GO TO 7590 :: REM GO TO 11
7570 REM STATEMENT 10
7580 WORK(II)=0.0
7590 NEXT II
7600 REM
7610 REM THE INVERSE ITERATION IS PERFORMED ON THE MATRIX TIL THE NORM ON RHS GT
ER THAN BOUND
7620 BOUND=0.01/(EX*N)
7630 NS=0
7640 ITER=1
7650 REM
7660 REM THE BACKSUBSTITUTION
7670 REM
7680 REM STATEMENT 12
7690 R=0.0
7700 FOR II=1 TO M
7710 J=M-II+1
7720 S=WORK(J)
7730 IF J=M THEN 7800 :: REM GO TO 14
7740 L=J+1
7750 FOR K=L TO M
7760 SR=WORK(K)
7770 REM STATEMENT 13
7780 S=S-SR*A(J,K)
7790 NEXT K
7800 REM STATEMENT 14
7810 WORK(J)=S/A(J,J)
7820 T=ABS(WORK(J))
7830 IF R>=T THEN 7850 :: REM GO TO 15
7840 R=T
7850 REM STATEMENT 15
7855 NEXT II
7860 REM
7870 REM REM THE COMPUTATION OF THE RIGHT-HAND SIDE VECTOR FOR THE ITERATION STE
P.
7880 REM
7890 FOR II=1 TO M
7900 WORK(II)=WORK(II)/R
7910 NEXT II
7920 REM
7930 REM THE COMPUTATION OF RESIDUALS
7940 REM
7950 R1=0.0
7960 FOR II=1 TO M
7970 T=0.0
7980 FOR J=II TO M
7990 T=T+A(II,J)*WORK(J)
8000 NEXT J
```

```
8010 T=ABS(T)
8020 IF R1>=T THEN 8040 :: REM GO TO 18
8030 R1=T
8040 REM STATEMENT 18
8050 NEXT II
8060 IF ITER=1 THEN 8080 :: REM GO TO 19
8070 IF PREVIS<=R1 THEN 8340 :: REM GO TO 24
8080 REM STATEMENT 19
8090 FOR II=1 TO M
8100 REM STATEMENT 20
8110 VECR(II,IVEC)=WORK(II)
8120 NEXT II
8130 PREVIS=R1
8140 IF NS=1 THEN 8340 :: REM GO TO 2454
8150 IF ITER>40 THEN 8360 :: REM GO TO 25
8160 ITER=ITER+1
8170 IF R<BOUND THEN 8200 :: REM GO TO 21
8180 NS=1
8190 REM REM GAUSSIAN ELIMINATION OF THE RIGHT-HAND SIDE VECTOR
8200 REM STATEMENT 21
8210 K=M-1
8220 FOR II=1 TO K
8230 R=WORK(II+1)
8240 IF IWORK(II)=0 THEN 8280 :: REM GO TO 22
8250 WORK(II+1)=WORK(II)+WORK(II+1)*A(II+1,II)
8260 WORK(II)=R
8270 GO TO 8300 :: REM GO TO 23
8280 REM STATEMENT 22
8290 WORK(II+1)=WORK(II+1)+WORK(II)*A(II+1,II)
8300 REM STATEMENT 23
8310 NEXT II
8320 GO TO 7680 :: REM GO TO 12
8330 REM
8340 REM STATEMENT 24
8350 INDIC(IVEC)=2
8360 REM STATEMENT 25
8370 IF M=N THEN 8420 :: REM GO TO 27
8380 J=M+1
8390 FOR II=J TO N
8400 VECR(II,IVEC)=0.0
8410 NEXT II
8420 REM STATEMENT 27
8430 SUBEND
```

```
9000 SUB COMPVE(N,NM,M,IVEC,A(,),VECR(,),H(,),EVR(),EVI(),INDIC(),IWORK(),SUBDIA
(),WORK1(),WORK2(),WORK(),EPS,EX)
9010 REM
9020 REM THIS SUBROUTINE FINDS THE IMAGINARY EIGENVALUES OF THE MATIX
9030 REM
9040 FKSI=EVR(IVEC)
9050 ETA=EVI(IVEC)
9060 IF IVEC=M THEN 9180 :: REM GO TO 2
9070 K=IVEC+1
9080 R=0.0
9090 FOR II=K TO M
9100 IF FKSI<>EVR(II)THEN 9130 :: REM GO TO 1
9110 IF ABS(ETA)<>ABS(EVI(II))THEN 9130 :: REM GO TO 1
9120 R=R+3.0
9130 REM STATEMENT 1
9140 NEXT II
9150 R=R*EX
9160 FKSI=FKSI+R
9170 ETA=ETA+R
9180 REM STATEMENT 2
9190 R=FKSI*FKSI+ETA*ETA
9200 S=2.0*FKSI
9210 L=M-1
9220 FOR II=1 TO M
9230 FOR J=II TO M
9240 D=0.0
9250 A(J,II)=0.0
9260 FOR K=II TO J
9270 REM STATEMENT 3
9280 D=D+H(II,K)*H(K,J)
9290 NEXT K
9300 REM STATEMENT 4
9310 A(II,J)=D-S*H(II,J)
9320 NEXT J
9330 REM STATEMENT 5
9340 A(II,II)=A(II,II)+R
9350 NEXT II
9360 FOR II=1 TO L
9370 R=SUBDIA(II)
9380 A(II+1,II)=-S*R
9390 I1=II+1
9400 FOR J=1 TO I1
9410 A(J,II)=A(J,II)+R*H(J,II+1)
9420 NEXT J
9430 IF II=1 THEN 9450 :: REM GO TO 7
9440 A(II+1,II-1)=R*SUBDIA(II-1)
9450 REM STATEMENT 7
9460 FOR J=II TO M
9470 REM STATEMENT 8
9480 A(II+1,J)=A(II+1,J)+R*H(II,J)
9490 NEXT J
9500 REM STATEMENT 9
9510 NEXT II
```

```
9520 REM REM GAUSSIAN ELIMINATION ROUTINE
9530 REM
9540 K=M-1
9550 FOR II=1 TO K
9560 I1=II+1
9570 I2=II+2
9580 IWORK(II)=0
9590 IF II=K THEN 9610 :: REM GO TO 10
9600 IF A(II+2,II)<>0.0 THEN 9660 :: REM GO TO 11
9610 REM STATEMENT 10
9620 IF A(II+1,II)<>0.0 THEN 9660 :: REM GO TO 11
9630 IF A(II,II)<>0.0 THEN 9950 :: REM GO TO 18
9640 A(II,II)=EPS
9650 GO TO 9950 :: REM GO TO 18
9660 REM STATEMENT 11
9670 IF II=K THEN 9730 :: REM GO TO 12
9680 IF ABS(A(II+1,II))>=ABS(A(II+2,II))THEN 9730 :: REM GO TO 12
9690 IF ABS(A(II,II))>=ABS(A(II+2,II))THEN 9870 :: REM GO TO 16
9700 L=II+2
9710 IWORK(II)=2
9720 GO TO 9770 :: REM GO TO 13
9730 REM STATEMENT 12
9740 IF ABS(A(II,II))>=ABS(A(II+1,II))THEN 9840 :: REM GO TO 15
9750 L=II+1
9760 IWORK(II)=1
9770 REM STATEMENT 13
9780 FOR J=II TO M
9790 R=A(II,J)
9800 A(II,J)=A(L,J)
9810 REM STATEMENT 14
9820 A(L,J)=R
9830 NEXT J
9840 REM STATEMENT 15
9850 IF II<>K THEN 9870 :: REM GO TO 16
9860 I2=I1
9870 REM STATEMENT 16
9880 FOR L=I1 TO I2
9890 R=-A(L,II)/A(II,II)
9900 A(L,II)=R
9910 FOR J=I1 TO M
9920 REM STATEMENT 17
9930 A(L,J)=A(L,J)+R*A(II,J)
9940 NEXT J
9945 NEXT L
9950 REM STATEMENT 18
9960 NEXT II
9970 IF A(M,M)<>0.0 THEN 9990 :: REM GO TO 19
9980 A(M,M)=EPS
9990 REM STATEMENT 19
10000 FOR II=1 TO N
10010 IF II>M THEN 10050 :: REM GO TO 20
10020 VECR(II,IVEC)=1.0
10030 VECR(II,IVEC-1)=1.0
```

```
10040 GO TO 10080 :: REM GO TO 21
10050 REM STATEMENT 20
10060 VECR(II,IVEC)=0.0
10070 VECR(II,IVEC-1)=0.0
10080 REM STATEMENT 21
10090 NEXT II
10100 REM REM INVERSE ITERATION IS PERFORMED
10110 REM
10120 BOUND=0.01/(EX*N)
10130 NS=0
10140 ITER=1
10150 FOR II=1 TO M
10160 WORK(II)=H(II,II)-FKSI
10170 NEXT II
10180 REM
10190 REM SEQUENCE OF COMPLEX VECTORS
10200 REM
10210 REM STATEMENT 23
10220 FOR II=1 TO M
10230 D=WORK(II)*VECR(II,IVEC)
10240 IF II=1 THEN 10260 :: REM GO TO 24
10250 D=D+SUBDIA(II-1)*VECR(II-1,IVEC)
10260 REM STATEMENT 24
10270 L=II+1
10280 IF L>M THEN 10320 :: REM GO TO 26
10290 FOR K=L TO M
10300 D=D+H(II,K)*VECR(K,IVEC)
10310 NEXT K
10320 REM STATEMENT 26
10330 VECR(II,IVEC-1)=D-ETA*VECR(II,IVEC-1)
10340 REM STATEMENT 27
10350 NEXT II
10360 REM
10370 REM GAUSSIAN ELIMINATION OF RIGHT HAND SIDE VECTOR
10380 REM
10390 K=M-1
10400 FOR II=1 TO K
10410 L=II+IWORK(II)
10420 R=VECR(L,IVEC-1)
10430 VECR(L,IVEC-1)=VECR(II,IVEC-1)
10440 VECR(II,IVEC-1)=R
10450 VECR(II+1,IVEC-1)=VECR(II+1,IVEC-1)+A(II+1,II)*R
10460 IF II=K THEN 10480 :: REM GO TO 28
10470 VECR(II+2,IVEC-1)=VECR(II+2,IVEC-1)+A(II+2,II)*R
10480 REM STATEMENT 28
10490 NEXT II
10500 REM
10510 REM THE COMPUTATION OF THE REAL PART
10520 REM
10530 FOR II=1 TO M
10540 J=M-II+1
10550 D=VECR(J,IVEC-1)
10560 IF J=M THEN 10620 :: REM GO TO 30
```

```
10570 L=J+1
10580 FOR K=L TO M
10590 D1=A(J,K)
10600 D=D-D1*VECR(K,IVEC-1)
10610 NEXT K
10620 REM STATEMENT 30
10630 VECR(J,IVEC-1)=D/A(J,J)
10640 REM STATEMENT 31
10650 NEXT II
10660 REM
10670 REM THE COMPUTATION OF THE IMAGINARY PART
10680 REM
10690 FOR II=1 TO M
10700 D=WORK(II)*VECR(II,IVEC-1)
10710 IF II=1 THEN 10730 :: REM GO TO 32
10720 D=D+SUBDIA(II-1)*VECR(II-1,IVEC-1)
10730 REM STATEMENT 32
10740 L=II+1
10750 IF L>M THEN 10790 :: REM GO TO 34
10760 FOR K=L TO M
10770 D=D+H(II,K)*VECR(K,IVEC-1)
10780 NEXT K
10790 REM STATEMENT 34
10800 VECR(II,IVEC)=(VECR(II,IVEC)-D)/ETA
10810 REM STATEMENT 35
10820 NEXT II
10830 REM
10840 L=1
10850 S=0.0
10860 FOR II=1 TO M
10870 R=VECR(II,IVEC)^2+VECR(II,IVEC-1)^2
10880 IF R<=S THEN 10910 :: REM GO TO 36
10890 S=R
10900 L=II
10910 REM STATEMENT 36
10920 NEXT II
10930 U=VECR(L,IVEC-1)
10940 V=VECR(L,IVEC)
10950 FOR II=1 TO M
10960 B=VECR(II,IVEC)
10970 R=VECR(II,IVEC-1)
10980 VECR(II,IVEC)=(R*U+B*V)/S
10990 VECR(II,IVEC-1)=(B*U-R*V)/S
11000 NEXT II
11010 REM
11020 REM THE COMPUTATION OF THE RESIDUALS
11030 REM
11040 B=0.0
11050 FOR II=1 TO M
11060 R=WORK(II)*VECR(II,IVEC-1)-ETA*VECR(II,IVEC)
11070 U=WORK(II)*VECR(II,IVEC)+ETA*VECR(II,IVEC-1)
11080 IF II=1 THEN 11110 :: REM GO TO 38
```

```
11090 R=R+SUBDIA(II-1)*VECR(II-1,IVEC-1)
11100 U=U+SUBDIA(II-1)*VECR(II-1,IVEC)
11110 REM STATEMENT 38
11120 L=II+1
11130 IF L>M THEN 11180 :: REM GO TO 40
11140 FOR J=L TO M
11150 R=R+H(II,J)*VECR(J,IVEC-1)
11160 U=U+H(II,J)*VECR(J,IVEC)
11170 NEXT J
11180 REM STATEMENT 40
11190 U=R*R+U*U
11200 IF B>=U THEN 11220 :: REM GO TO 41
11210 B=U
11220 REM STATEMENT 41
11230 NEXT II
11240 IF ITER=1 THEN 11260 :: REM GO TO 42
11250 IF PREVIS<=B THEN 11400 :: REM GO TO 44
11260 REM STATEMENT 42
11270 FOR II=1 TO N
11280 WORK1(II)=VECR(II,IVEC)
11290 REM STATEMENT 43
11300 WORK2(II)=VECR(II,IVEC-1)
11310 NEXT II
11320 PREVIS=B
11330 IF NS=1 THEN 11450 :: REM GO TO 46
11340 IF ITER>6 THEN 11480 :: REM GO TO 47
11350 ITER=ITER+1
11360 IF BOUND>SQR(S)THEN 10210 :: REM GO TO 23
11370 NS=1
11380 GO TO 10210 :: REM GO TO 23
11390 REM
11400 REM STATEMENT 44
11410 FOR II=1 TO N
11420 VECR(II,IVEC)=WORK1(II)
11430 VECR(II,IVEC-1)=WORK2(II)
11440 NEXT II
11450 REM STATEMENT 46
11460 INDIC(IVEC-1)=2
11470 INDIC(IVEC)=2
11480 REM STATEMENT 47
11490 SUBEND
```

```
12000 SUB INPUTT(ROW,COL,WORK(,),IERROR)
12010 REM INPUT IN A MATRIX
12020 REM
12030 PRINT
12040 INPUT "ENTER #ROWS,#COLS OF MATRIX-":ROW,COL
12050 REM
12060 REM
12070 REM
12080 FOR I=1 TO ROW
12090 PRINT "ROW #",I
12100 FOR J=1 TO COL
12110 PRINT J,"  "
12120 INPUT WORK(I,J)
12130 NEXT J
12140 NEXT I
12150 SUBEND
```

APPENDIX G

```
10 REM PROGRAM TO CALCULATE F MATRIX
15 REM
20 REM BY A
25 REM
100 REM 4TH ORDER RUNGA-KUTTA ROUTINE FOR UP TO AN 8TH ORDER SYSTEM
110 REM
120 CALL CLEAR
130 REM
140 OPEN #1:"PIO"
150 REM
160 DIM X(8,8),XWORK(8,8),XNEW(8,8),AMAT(8,8)
170 DIM F1(8,8),F2(8,8),F3(8,8),F4(8,8),F(8,8)
180 DIM K1(8,8),K2(8,8),K3(8,8),K4(8,8)
185 DIM JJ(8,8),OUT(8,8),A(8,8),B(8,8),FWORK1(8,8),FWORK2(8,8)
190 REM
200 REM INITIALIZE ALL VARIABLES
210 REM
220 ANGLE=0.0
230 FOR I=1 TO 8
240 FOR J=1 TO 8
250 X(I,J)=0.0
270 NEXT J
280 NEXT I
290 REM
300 CALL CLEAR
310 REM
320 REM ENTER THE  INPUT PARAMETERS
330 REM
340 INPUT "ENTER # OF INCREMENTS":INCR
350 DELTANGLE=2*PI/INCR
360 REM
370 INPUT "ENTER ORDER OF SYSTEM":ORDER
372 INPUT "ENTER IN B:":BE
374 INPUT "ENTER IN MU:":MU
376 INPUT "ENTER IN GAMMA:":GAMMA
378 INPUT "ENTER IN PSQUARE:":PSQUARE
379 PRINT "INPUT J MATRIX"
380 CALL INPUTT(ORDER,ORDER,JJ(,),IERROR)
381 CALL CLEAR :: PRINT "ENTER F(0) MATRIX"
382 CALL INPUTT(ORDER,ORDER,X(,),IERROR)
383 CALL CLEAR
384 REM
385 OPEN #2:"DSK.DATATWO.FFMATRIX",SEQUENTIAL,DISPLAY ,UPDATE,FIXED
386 PRINT #2:INCR+1
387 PRINT #2:BE :: PRINT #2:MU :: PRINT #2:GAMMA :: PRINT #2:PSQUARE
388 PRINT #2:JJ(1,1):: PRINT #2:JJ(1,2):: PRINT #2:JJ(2,1):: PRINT #2:JJ(2,2)
389 PRINT #2:ANGLE :: PRINT #2:X(1,1):: PRINT #2:X(1,2):: PRINT #2:X(2,1):: PRIN
T #2:X(2,2)
```

```
390 REM MASTER LOOP
400 REM
410 FOR N=1 TO INCR STEP 1
420 DISPLAY AT(2,4):"N=",N
430 REM
440 REM FIRST RUNGA-KUTTA CONSTANT
450 REM
460 CALL FCNN(ANGLE,X(,),JJ(,),F1(,),BE,MU,GAMMA,PSQUARE)
470 REM
480 FOR I=1 TO ORDER
490 FOR J=1 TO ORDER
500 K1(I,J)=DELTANGLE*F1(I,J)
530 REM
560 XWORK(I,J)=X(I,J)+0.5*K1(I,J)
570 NEXT J
580 NEXT I
590 REM
600 REM SECOND RUNGA-KUTTA CONSTANT
610 REM
620 CALL FCNN(ANGLE+0.5*DELTANGLE,XWORK(,),JJ(,),F2(,),BE,M
630 REM
640 FOR I=1 TO ORDER
650 FOR J=1 TO ORDER
660 K2(I,J)=DELTANGLE*F2(I,J)
690 REM
720 XWORK(I,J)=X(I,J)+0.5*K2(I,J)
730 NEXT J
740 NEXT I
750 REM
760 REM THIRD RUNGA-KUTTA CONSTANT
770 REM
780 CALL FCNN(ANGLE+0.5*DELTANGLE,XWORK(,),JJ(,),F3(,),BE,M
790 REM
800 FOR I=1 TO ORDER
810 FOR J=1 TO ORDER
820 K3(I,J)=DELTANGLE*F3(I,J)
850 REM
880 XWORK(I,J)=X(I,J)+K3(I,J)
890 NEXT J
900 NEXT I
910 REM
920 REM FOURTH RUNGA-KUTTA CONSTANT
930 REM
940 CALL FCNN(ANGLE+DELTANGLE,XWORK(,),JJ(,),F4(,),BE,ML
950 REM
960 FOR I=1 TO ORDER
970 FOR J=1 TO ORDER
980 K4(I,J)=DELTANGLE*F4(I,J)
990 NEXT J
1000 NEXT I
1010 REM
```

```
1020 REM EVALUATION OF STATE VARIABLES
1030 REM
1040 FOR I=1 TO ORDER
1050 FOR J=1 TO ORDER
1060 XNEW(I,J)=X(I,J)+(1/6)*(K1(I,J)+2*K2(I,J)+2*K3(I,J)+K4
1070 NEXT J
1080 NEXT I
1090 REM
1100 REM UPDATE STATE VARIABLES
1110 REM
1115 REM PRINT #1:"     ANGLE=",(ANGLE+DELTANGLE)*180/(PI)
1116 IF (N/10)=INT(N/10)THEN 1117 ELSE 1120
1117 PRINT #2:ANGLE+DELTANGLE
1120 FOR I=1 TO ORDER
1130 FOR J=1 TO ORDER
1140 X(I,J)=XNEW(I,J)
1145 REM PRINT #1,USING "   F(#,#)=###.#####^^^^":I,J,XNEW(I,J)
1150 NEXT J
1160 NEXT I
1162 IF (N/10)=INT(N/10)THEN 1165 ELSE 1170
1165 PRINT #2:X(1,1):: PRINT #2:X(1,2):: PRINT #2:X(2,1):: PRINT
1170 REM
1180 ANGLE=ANGLE+DELTANGLE
1190 REM
1200 NEXT N
1210 REM
1220 REM PRINT OUTPUT
1230 REM
1231 PRINT #1:"-----------------------------------------------
---"
1232 PRINT #1
1233 PRINT #1,USING "  BE=##.##^^^^   MU=##.##^^^^":BE,MU
1234 PRINT #1
1235 PRINT #1,USING "  GAMMA=##.##^^^^   PSQUARE=##.##^^^^":GAMMA
1236 PRINT #1
1237 PRINT #1
1240 FOR I=1 TO ORDER
1250 FOR J=1 TO ORDER
1260 PRINT #1,USING "    X(#,#)=###.#####^^^^":I,J,X(I,J)
1270 NEXT J
1280 NEXT I
1290 PRINT #1
1300 PRINT #1
1310 PRINT #1
1320 REM
1325 REM NEXT GAMMA
1326 REM
1330 CLOSE #1
1335 CLOSE #2
1340 REM
1350 END
```

```
1360 SUB FCNN(T,XWORK(,),JJ(,),F(,),BE,MU,GAMMA,PSQUARE)
1370 REM
1380 CALL MATRIXA(T,AMAT(,),BE,MU,GAMMA,PSQUARE)
1390 REM
1400 CALL MATRIXMULT(AMAT(,),XWORK(,),2,2,2,FWORK1(,))
1401 REM
1402 CALL MATRIXMULT(XWORK(,),JJ(,),2,2,2,FWORK2(,))
1403 REM
1404 CALL MATRIXADD(FWORK1(,),FWORK2(,),2,2,F(,),-1)
1406 REM
1410 REM
1420 REM
1430 SUBEND
```

```
2000 SUB MATRIXA(PSI,MATA(,),BE,MU,GAMMA,PSQUARE)
2001 REM
2002 REM ALGORITHM TO CALCULATE STATE TRANSITION MATRIX
2003 REM FOR A PERIODIC FUNCTION
2004 REM IN PARTICULAR-HELICOPTER BLADE, NO FEEDBACK
2005 REM  UPDATED 16 AUG 1984
2006 REM
2007 DEF ARCSIN(ZZ)=ATN(ZZ/SQR(1-ZZ*ZZ))
2008 PSIT=PSI
2010 IF PSIT>=0.0 AND PSIT<PI THEN 2020 ELSE 2012
2012 IF PSIT>=PI AND PSIT<(PI+ARCSIN(BE/MU))THEN 2030 ELSE 2014
2014 IF PSIT>=(2*PI-ARCSIN(BE/MU))AND PSIT<(2*PI)-.000001 THEN 2030 ELSE 2016
2016 IF PSIT>(2*PI)-.000001 THEN PSIT=0.0 :: GO TO 2020 :: ELSE GO TO 2018
2018 GO TO 2040
2020 REM PRINT #1:"2020"
2022 K=(1/3)*BE^3*MU*COS(PSIT)+(1/4)*BE^2*MU^2*SIN(2*PSIT)
2023 REM
2024 C=(1/4)*BE^4+(1/3)*BE^3*MU*SIN(PSIT)
2025 REM
2026 GO TO 2050
2030 REM PRINT #1:"2030"
2031 REM
2032 K=(1/3)*BE^3*MU*COS(PSIT)+(1/4)*BE^2*MU^2*SIN(2*PSIT)+MU^4*(-(1/12)*SIN(2*P
SIT)+(1/24)*SIN(4*PSIT))
2033 REM
2034 C=(1/4)*BE^4+(1/3)*BE^3*MU*SIN(PSIT)+MU^4*((1/16)-(1/12)*COS(2*PSIT)+(1/48)
*COS(4*PSIT))
2035 REM
2036 GO TO 2050
2040 REM PRINT #1:"2040"
2041 REM
2042 K=-(1/3)*BE^3*MU*COS(PSIT)-(1/4)*BE^2*MU^2*SIN(2*PSIT)
2043 REM
2044 C=-(1/4)*BE^4-(1/3)*BE^3*MU*SIN(PSIT)
2045 REM
2050 REM CONSTRUCT MATRIX A
2051 REM
2052 MATA(1,1)=0
2053 MATA(1,2)=1
2054 MATA(2,1)=-(GAMMA/2)*(2*PSQUARE/GAMMA+K)
2055 MATA(2,2)=-(GAMMA*C/2)
2056 REM
2057 REM RETURN TO MAIN PROGRAM
2060 SUBEND
```

```
4000 SUB INPUTT(ROW,COL,WORK(,),IERROR)
4010 REM  INPUT IN A MATRIX DATA
4020 REM
4030 PRINT
4040 INPUT "ENTER #ROWS,#COLS OF MATRIX-":ROW,COL
4050 IF ROW>8 THEN 4040
4060 IF ROW<2 THEN IERROR=1
4080 FOR I=1 TO ROW
4090 PRINT "ROW #",I
4100 FOR J=1 TO COL
4110 PRINT J,"  "
4120 INPUT WORK(I,J)
4130 NEXT J
4150 NEXT I
4160 SUBEND :: REM FROM THE SUBROUTINE
```

```
7000 SUB MATRIXMULT(A(,),B(,),ROWA,COLA,COLB,OUT(,))
7001 REM
7002 REM MATRIX MULTIPLICATION ROUTINE
7003 REM INPUTS    MATRIX "A", MATRIX "B".DIMENSION OF AAND COLUMN DIMENSION OF
B
7004 FOR I=1 TO ROWA
7005 FOR J=1 TO COLA
7006 OUT(I,J)=0.0
7007 NEXT J
7008 NEXT I
7009 REM
7010 FOR I=1 TO ROWA
7020 FOR K=1 TO COLB
7030 FOR J=1 TO COLA
7040 SUM=A(I,J)*B(J,K)
7050 OUT(I,K)=OUT(I,K)+SUM
7060 NEXT J
7070 NEXT K
7080 NEXT I
7090 REM
7999 SUBEND :: REM RETURN TO MAIN PROGRAM
```

```
9000 SUB MATRIXADD(A(,),B(,),ROWA,COLA,OUT(,),SIGNN)
9010 REM
9020 REM MATRIX ADDITION/SUBTRACTION ROUTINE
9030 REM
9040 REM SIGNN: +1-ADDITION,-1-SUBTRACTION
9050 REM
9060 FOR I=1 TO ROWA
9070 FOR J=1 TO COLA
9080 IF SIGNN=-1 THEN 9100
9090 OUT(I,J)=A(I,J)+B(I,J)
9095 GO TO 9110
9100 OUT(I,J)=A(I,J)-B(I,J)
9110 NEXT J
9120 NEXT I
9130 REM
9140 SUBEND
```

APPENDIX H

```
100 REM FOURIER COEFFICIENT ANALYSIS ROUTINE
110 REM WRITTEN IN TI EXTENDED BASIC
120 REM
125 CALL CLEAR
130 REM PROGRAM READS TAPE#2 INFORMATION FROM F-MATRIX PROGRAM
140 REM AND DERIVES 18 FOURIER COEFFICIENTS AND PRINTS OUTPUT
150 REM
160 DIM Y(2,2),A(19,2,2),B(19,2,2)
170 REM
171 FOR K=1 TO 19
172 FOR I=1 TO 2
173 FOR J=1 TO 2
174 A(K,I,J)=0.0
175 B(K,I,J)=0.0
176 NEXT J
177 NEXT I
178 NEXT K
180 OPEN #1:"PIO"
190 OPEN #2:"DSK.DATATWO.FFMATRIX",SEQUENTIAL,DISPLAY ,UPDATE,FIXED
200 REM
210 INPUT #2:INCR
220 PRINT #1:"         INCREMENT OF RUN EQUALS ";INCR-1;" STEPS"
230 REM
240 INPUT #2:BE,MU,GAMMA,PSQUARE
250 PRINT #1
260 PRINT #1:"     FOR THIS RUN;"
270 PRINT #1,USING "   B=##.##      MU=##.##      GAMMA=##.##      PSQUARE=##.##":B
E,MU,GAMMA,PSQUARE
280 REM
290 INPUT #2:LAMBDA1,DUMP,DUMP,LAMBDA2
300 PRINT #1
310 PRINT #1:"                        POINCARE EXPONENTS"
320 PRINT #1
330 PRINT #1,USING "         OMEGA1=###.#####^^^^           OMEGA2=###.#####^^^^
":LAMBDA1,LAMBDA2
340 REM
350 PRINT #1
360 PRINT #1:"                        SYSTEM OF EIGENVECTORS"
370 PRINT #1
380 REM
390 REM
400 IMAGE ###.####^^^^          ###.####^^^^
402 N=0.0
410 INPUT #2:ANGLE
420 INPUT #2:Y(1,1),Y(1,2),Y(2,1),Y(2,2)
430 PRINT #1,USING 400:Y(1,1),Y(1,2)
440 PRINT #1,USING 400:Y(2,1),Y(2,2)
442 REM
```

```
445 CALL FOURIER(ANGLE,Y(,),A(,,),B(,,),N)
446 REM
450 REM
460 REM
470 FOR N=1 TO 36
480 INPUT #2:X
490 INPUT #2:Y(1,1),Y(1,2),Y(2,1),Y(2,2)
500 CALL FOURIER(X,Y(,),A(,,),B(,,),N)
510 NEXT N
520 REM
530 REM
540 FOR K=1 TO 19
550 FOR I=1 TO 2
560 FOR J=1 TO 2
570 A(K,I,J)=(2/360)*A(K,I,J)*(10/3)
580 B(K,I,J)=(2/360)*B(K,I,J)*(10/3)
590 NEXT J
600 NEXT I
610 NEXT K
620 REM
630 REM PRINT OUT RESULTS
640 FOR N=1 TO 19
650 PRINT #1
660 PRINT #1,USING "           A## COEFFICIENT":N-1
670 PRINT #1
680 PRINT #1,USING 400:A(N,1,1),A(N,1,2)
690 PRINT #1,USING 400:A(N,2,1),A(N,2,2)
700 PRINT #1
710 PRINT #1,USING "           B## COEFFICIENT":N-1
720 PRINT #1
730 PRINT #1,USING 400:B(N,1,1),B(N,1,2)
740 PRINT #1,USING 400:B(N,2,1),B(N,2,2)
750 NEXT N
760 REM
770 REM
780 PRINT #1
790 PRINT #1
800 PRINT #1
810 CLOSE #1
820 CLOSE #2
830 REM
840 END
```

```
880 SUB FOURIER(X,Y(,),A(,,),B(,,),N)
890 REM
900 REM  FOURIER COEFFICIENT DETERMINATION
910 REM
912 IF N=0. THEN CONSTANT=1 ELSE GO TO 914
913 GO TO 920
914 IF N=36 THEN CONSTANT=1 ELSE GO TO 916
915 GO TO 920
916 IF INT(N/2)=N/2 THEN CONSTANT=2 ELSE CONSTANT=4
918 REM
920 FOR KK=1 TO 19
930 FOR II=1 TO 2
940 FOR JJ=1 TO 2
945 DISPLAY AT(5,5):N,CONSTANT,Y(1,1)
950 REM
960 A(KK,II,JJ)=A(KK,II,JJ)+CONSTANT*Y(II,JJ)*COS((KK-1)*X)
970 REM
980 B(KK,II,JJ)=B(KK,II,JJ)+CONSTANT*Y(II,JJ)*SIN((KK-1)*X)
990 REM
1000 NEXT JJ
1010 NEXT II
1020 NEXT KK
1030 REM
1040 SUBEND
```

APPENDIX I

```
100 REM FOURIER COEFFICIENT ANALYSIS ROUTINE
110 REM WRITTEN IN TI EXTENDED BASIC
120 REM
125 CALL CLEAR
130 REM PROGRAM READS TAPE#2 INFORMATION FROM F-MATRIX PROGRAM
140 REM AND DERIVES 18 FOURIER COEFFICIENTS AND PRINTS OUTPUT
145 REM FOR GAIN FOURIER SERIES OF 3 [B] MATRICES
146 REM
150 REM
160 DIM Y(2,2),A(2,2),B(2,2),G(2,2)
161 DIM AM(2,2),BM(2,2),I1(2,2),I2(2,2),OUT(2,2)
162 DIM AG1(19,2,2),AG2(19,2,2),AG3(19,2,2),BG1(19,2,2),BG2(19,2,2),BG3(19,2,2) ;
163 DIM B1(2,1),B2(2,1),B3(2,1),G1(4,4),G2(4,4),G3(4,4)
164 DIM YINV(4,4),AF(19,2,2),BF(19,2,2)
165 REM
170 REM
171 FOR K=1 TO 19
172 FOR I=1 TO 2
173 FOR J=1 TO 2
174 AG1(K,I,J)=0.0 :: AG2(K,I,J)=0.0 :: AG3(K,I,J)=0.0
175 BG1(K,I,J)=0.0 :: BG2(K,I,J)=0.0 :: BG3(K,I,J)=0.0
176 NEXT J
177 NEXT I
178 NEXT K
180 OPEN #1:"PIO"
190 OPEN #2:"DSK.DATATWO.FFMATRIX",SEQUENTIAL,DISPLAY ,UPDATE,FIXED
200 REM
202 B1(1,1)=1.
203 B1(2,1)=0.
204 B2(1,1)=0.
205 B2(2,1)=1.
206 B3(1,1)=1.
207 B3(2,1)=1.
208 REM
210 INPUT #2:INCR
220 PRINT #1:"        INCREMENT OF RUN EQUALS ";INCR-1;" STEPS"
230 REM
240 INPUT #2:BE,MU,GAMMA,PSQUARE
250 PRINT #1
260 PRINT #1:"     FOR THIS RUN;"
270 PRINT #1,USING "   B=##.##      MU=##.##      GAMMA=##.##      PSQUARE=##.##":B
E,MU,GAMMA,PSQUARE
280 REM
290 INPUT #2:LAMBDA1,DUMP,DUMP,LAMBDA2
300 PRINT #1
310 PRINT #1:"                        POINCARE EXPONENTS"
```

```
320 PRINT #1
330 PRINT #1,USING "          OMEGA1=###.#####^^^^          OMEGA2=###.#####^^^^
":LAMBDA1,LAMBDA2
340 REM
350 PRINT #1
360 PRINT #1:"                    SYSTEM OF EIGENVECTORS"
370 PRINT #1
380 REM
390 REM
400 IMAGE ###.####^^^^          ###.####^^^^
401 N=0.0
402 FOR I=1 TO 2
403 FOR J=1 TO 2
404 G1(I,J)=0.0
405 G2(I,J)=0.0
406 G3(I,J)=0.0
407 NEXT J
408 NEXT I
409 REM
410 INPUT #2:ANGLE
420 INPUT #2:Y(1,1),Y(1,2),Y(2,1),Y(2,2)
430 PRINT #1,USING 400:Y(1,1),Y(1,2)
440 PRINT #1,USING 400:Y(2,1),Y(2,2)
441 REM
442 CALL INVERSE(2,Y(,),YINV(,))
443 CALL MATRIXMULT(YINV(,),B1(,),2,2,1,G1(,))
444 CALL MATRIXMULT(YINV(,),B2(,),2,2,1,G2(,))
445 CALL MATRIXMULT(YINV(,),B3(,),2,2,1,G3(,))
446 REM
447 CALL FOURIER(ANGLE,G1(,),AG1(,,),BG1(,,),N)
448 CALL FOURIER(ANGLE,G2(,),AG2(,,),BG2(,,),N)
449 CALL FOURIER(ANGLE,G3(,),AG3(,,),BG3(,,),N)
450 REM
460 REM
470 FOR N=1 TO 36
480 INPUT #2:X
490 INPUT #2:Y(1,1),Y(1,2),Y(2,1),Y(2,2)
491 REM
492 CALL INVERSE(2,Y(,),YINV(,))
493 CALL MATRIXMULT(YINV(,),B1(,),2,2,1,G1(,))
494 CALL MATRIXMULT(YINV(,),B2(,),2,2,1,G2(,))
495 CALL MATRIXMULT(YINV(,),B3(,),2,2,1,G3(,))
496 REM
497 CALL FOURIER(X,G1(,),AG1(,,),BG1(,,),N)
498 CALL FOURIER(X,G2(,),AG2(,,),BG2(,,),N)
499 CALL FOURIER(X,G3(,),AG3(,,),BG3(,,),N)
500 REM
510 NEXT N
520 REM
530 REM
```

```
540 FOR K=1 TO 19
550 FOR I=1 TO 2
560 FOR J=1 TO 2
570 AG1(K,I,J)=(2/360)*AG1(K,I,J)*(10/3)
572 AG2(K,I,J)=(2/360)*AG2(K,I,J)*(10/3)
574 AG3(K,I,J)=(2/360)*AG3(K,I,J)*(10/3)
580 BG1(K,I,J)=(2/360)*BG1(K,I,J)*(10/3)
582 BG2(K,I,J)=(2/360)*BG2(K,I,J)*(10/3)
584 BG3(K,I,J)=(2/360)*BG3(K,I,J)*(10/3)
590 NEXT J
600 NEXT I
610 NEXT K
620 REM
630 REM PRINT OUT RESULTS
635 IMAGE ###.####^^^^     ###.####^^^^      ###.####^^^^
640 FOR N=1 TO 19
650 PRINT #1
660 PRINT #1,USING "          A## COEFFICIENT":N-1
670 PRINT #1
680 PRINT #1,USING 635:AG1(N,1,1),AG2(N,1,1),AG3(N,1,1)
690 PRINT #1,USING 635:AG1(N,2,1),AG2(N,2.1),AG3(N,2,1)
700 PRINT #1
710 PRINT #1,USING "          B## COEFFICIENT":N-1
720 PRINT #1
730 PRINT #1,USING 635:BG1(N,1,1),BG2(N,1,1),BG3(N,1,1)
740 PRINT #1,USING 635:BG1(N,2,1),BG2(N,2,1),BG3(N,2,1)
750 NEXT N
760 REM
770 REM
780 PRINT #1
790 PRINT #1
800 PRINT #1
810 CLOSE #1
820 CLOSE #2
830 REM
840 END
850 REM
860 REM ------------------------------------------------
870 REM
```

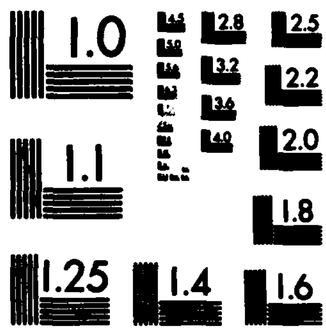
AD-A154 460 APPLICATION OF FLOQUET THEORY TO HELICOPTER BLADE FLAPPING STABILITY(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING J K MARCH DEC 84 AFIT/GAE/AA/84D-13 F/G 1/3 3/3

UNCLASSIFIED NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
880 SUB FOURIER(X,G(,),AF(,,),BF(,,),N)
890 REM
900 REM  FOURIER COEFFICIENT DETERMINATION
910 REM .
912 IF N=0. THEN CONSTANT=1 ELSE GO TO 914
913 GO TO 920
914 IF N=36 THEN CONSTANT=1 ELSE GO TO 916
915 GO TO 920
916 IF INT(N/2)=N/2 THEN CONSTANT=2 ELSE CONSTANT=4
918 REM
920 FOR KK=1 TO 19
930 FOR II=1 TO 2
940 FOR JJ=1 TO 2
945 DISPLAY AT(5,5):N
950 REM
960 AF(KK,II,JJ)=AF(KK,II,JJ)+CONSTANT*G(II,JJ)*COS((KK-1)*X)
970 REM
980 BF(KK,II,JJ)=BF(KK,II,JJ)+CONSTANT*G(II,JJ)*SIN((KK-1)*X)
990 REM
1000 NEXT JJ
1010 NEXT II
1020 NEXT KK
1030 REM
1040 SUBEND
```

```
5000 SUB INVERSE(N2,A(,),B(,))
5001 REM GAUSS-JORDAN MATRIX INVERSION ROUTINE
5010 REM
5011 REM     IDENTIFIERS
5013 REM A        A      COEFFICIENT MATRIX
5014 REM B        B      WORK MATRIX
5015 REM B1       BIG    BIGGEST VALUE
5017 REM D3       DETERM DETERMINANT
5018 REM E1       ERMES  ERROR FLAG
5019 REM H1       HOLD   WORK VARIABLE
5020 REM I2       INDEX  WORK MATRIX
5021 REM I3       IROW   ROW INDEX
5022 REM I4       ICOL   COLUMN INDEX
5023 REM I5       INVRS  PRINT-INVERSE FLAG
5024 REM N2       NCOL   NUMBER OF COLUMNS
5025 REM P1       PIVOT  PIVOT INDEX
5029 REM      END OF IDENTIFIERS
5080 REM
5090 E1=0 :: REM  BECOMES 1 FOR SINGULAR MATRIX
5100 I5=0 :: REM PRINT INVERSE MATRIX IF ZERO
5110 N3=1 :: REM NUMBER OF CONSTANT VECTORS
5120 FOR I=1 TO N2
5130 FOR J=1 TO N2
5140 B(I,J)=A(I,J)
5150 NEXT J
5170 I2(I,3)=0.
5180 NEXT I
5190 D3=1
5200 FOR I=1 TO N2
5210 REM
5220 REM  SEARCH FOR LARGEST PIVOT ELEMENT
5230 REM
5240 B1=0.
5250 FOR J=1 TO N2
5260 IF I2(J,3)=1 THEN 5350
5270 FOR K=1 TO N2
5280 IF I2(K,3)>1 THEN 6120
5290 IF I2(K,3)=1 THEN 5340
5300 IF B1>=ABS(B(J,K))THEN 5340
5310 I3=J
5320 I4=K
5330 B1=ABS(B(J,K))
5340 NEXT K
5350 NEXT J
5360 I2(I4,3)=I2(I4,3)+1
5370 I2(I,1)=I3
5380 I2(I,2)=I4
5390 REM  INTERCHANGE ROWS TO PUT PIVOT ON DIAGONAL
5400 IF I3=I4 THEN 5540
5410 D3=-D3
5420 FOR L=1 TO N2
5430 H1=B(I3,L)
```

```
5440 B(I3,L)=B(I4,L)
5450 B(I4,L)=H1
5460 NEXT L
5530 REM  DIVIDE PIVOT ROW BY PIVOT ELEMENT
5540 P1=B(I4,I4)
5550 D3=D3*P1
5560 B(I4,I4)=1
5570 FOR L=1 TO N2
5580 B(I4,L)=B(I4,L)/P1
5590 NEXT L
5640 REM
5650 REM  REDUCE NONPIVOT ROWS
5660 FOR L1=1 TO N2
5670 IF L1=I4 THEN 5770
5680 T=B(L1,I4)
5690 B(L1,I4)=0
5700 FOR L=1 TO N2
5710 B(L1,L)=B(L1,L)-B(I4,L)*T
5720 NEXT L
5770 NEXT L1
5780 NEXT I
5790 REM
5800 REM  INTERCHANGE COLUMNS
5810 REM
5820 FOR I=1 TO N2
5830 L=N2-I+1
5840 IF I2(L,1)=I2(L,2)THEN 5920
5850 I3=I2(L,1)
5860 I4=I2(L,2)
5870 FOR K=1 TO N2
5880 H1=B(K,I3)
5890 B(K,I3)=B(K,I4)
5900 B(K,I4)=H1
5910 NEXT K
5920 NEXT I
5930 FOR K=1 TO N2
5940 IF I2(K,3)<>1 THEN 6120
5950 NEXT K
5960 E1=0.
6000 GO TO 6110
6010 PRINT
6020 PRINT "   MATRIX INVERSION   "
6030 FOR I=1 TO N2
6040 FOR J=1 TO N2
6050 PRINT USING "##.###^^^^":B(I,J)
6060 NEXT J
6070 PRINT
6080 NEXT I
6085 BREAK
6090 PRINT
6100 PRINT "DETERMINANT =":D3
6110 SUBEND :: REM IF INVERSE IS PRINTED
```

```
7000 SUB MATRIXMULT(AM(,),BM(,),ROWA,COLA,COLB,OUT(,))
7001 REM
7002 REM MATRIX MULTIPLICATION ROUTINE
7003 REM INPUTS    MATRIX "A", MATRIX "B".DIMENSION OF AAND COLUMN DIMENSION OF
B
7004 FOR I=1 TO ROWA
7005 FOR J=1 TO COLA
7006 OUT(I,J)=0.0
7007 NEXT J
7008 NEXT I
7009 REM
7010 FOR I=1 TO ROWA
7020 FOR K=1 TO COLB
7030 FOR J=1 TO COLA
7040 SUM=AM(I,J)*BM(J,K)
7050 OUT(I,K)=OUT(I,K)+SUM
7060 NEXT J
7070 NEXT K
7080 NEXT I
7090 REM
7999 SUBEND :: REM RETURN TO MAIN PROGRAM
```

APPENDIX J

```
10 REM PROGRAM TO CALCULATE F MATRIX
15 REM
20 REM BY A FOURTH ORDER RUNGA-KUTTA
25 REM
100 REM 4TH ORDER RUNGA-KUTTA ROUTINE FOR UP TO AN 8TH ORDER SYSTEM
110 REM
120 CALL CLEAR
130 REM
140 OPEN #1:"PIO"
150 REM
160 DIM X(2,2),XWORK(2,2),XNEW(2,2),AMAT(2,2)
170 DIM F1(2,2),F2(2,2),F3(2,2),F4(2,2),F(2,2)
180 DIM K1(8,8),K2(8,8),K3(8,8),K4(8,8)
185 DIM JJ(2,2),OUT(2,2),A(2,2),B(2,2),FWORK1(2,2),FWORK2(2,2)
186 DIM FEEDBACK(2,2),XPRIME(2,2),XFWORK(2,2),APRIME(2,2)
188 DIM AADD(2,2),BADD(2,2),AM(2,2),BM(2,2),I1(2,2),I2(2,2)
189 DIM XFEEDBACK(2,2),K1FB(2,2),K2FB(2,2),K3FB(2,2),K4FB(2,2)
190 DIM FEED1(2,2),FEED2(2,2),FEED3(2,2),FEED4(2,2)
191 DIM FINVERSE(2,2)
200 REM INITIALIZE ALL VARIABLES
210 REM
220 ANGLE=0.0
230 FOR I=1 TO 2
240 FOR J=1 TO 2
250 X(I,J)=0.0
255 XFEEDBACK(I,J)=0.0
256 IF I=J THEN XFEEDBACK(I,J)=1.00
270 NEXT J
280 NEXT I
290 REM
300 CALL CLEAR
310 REM
320 REM ENTER THE  INPUT PARAMETERS
330 REM
340 INPUT "ENTER # OF INCREMENTS":INCR
350 DELTANGLE=2*PI/INCR
360 REM
370 INPUT "ENTER ORDER OF SYSTEM":ORDER
372 INPUT "ENTER IN B:":BE
374 INPUT "ENTER IN MU:":MU
376 INPUT "ENTER IN GAMMA:":GAMMA
378 INPUT "ENTER IN PSQUARE:":PSQUARE
379 PRINT "INPUT J MATRIX"
380 CALL INPUTT(ORDER,ORDER,JJ(,),IERROR)
381 CALL CLEAR :: PRINT "ENTER F(0) MATRIX"
382 CALL INPUTT(ORDER,ORDER,X(,),IERROR)
383 CALL CLEAR
```

```
384 REM
385 REM OPEN #2:"DSK.DATATWO.FFMATRIX",SEQUENTIAL,DISPLAY ,UPDATE,FIXED
386 REM PRINT #2:INCR+1
387 REM PRINT #2:BE :: PRINT #2:MU :: PRINT #2:GAMMA :: PRINT #2:PSQUARE
388 REM PRINT #2:JJ(1,1):: PRINT #2:JJ(1,2):: PRINT #2:JJ(2,1):: PRINT #2:JJ(2,2
)
389 REM PRINT #2:ANGLE :: PRINT #2:X(1,1):: PRINT #2:X(1,2):: PRINT #2:X(2,1)::
PRINT #2:X(2,2)
390 REM MASTER LOOP
400 REM
410 FOR N=1 TO INCR STEP 1
420 DISPLAY AT(2,4):"N=",N
430 REM
440 REM FIRST RUNGA-KUTTA CONSTANT
450 REM
460 CALL FCNN(ANGLE,X(,),XFEEDBACK(,),JJ(,),F1(,),FEED1(,),BE,MU,GAMMA,PSQUARE)
470 REM
480 FOR I=1 TO ORDER
490 FOR J=1 TO ORDER
500 K1(I,J)=DELTANGLE*F1(I,J)
510 K1FB(I,J)=DELTANGLE*FEED1(I,J)
530 REM
560 XWORK(I,J)=X(I,J)+0.5*K1(I,J)
565 XFWORK(I,J)=XFEEDBACK(I,J)+0.5*K1FB(I,J)
570 NEXT J
580 NEXT I
590 REM
600 REM SECOND RUNGA-KUTTA CONSTANT
610 REM
620 CALL FCNN(ANGLE+0.5*DELTANGLE,XWORK(,),XFWORK(,),JJ(,),F2(,),FEED2(,),BE,MU,
GAMMA,PSQUARE)
630 REM
640 FOR I=1 TO ORDER
650 FOR J=1 TO ORDER
660 K2(I,J)=DELTANGLE*F2(I,J)
670 K2FB(I,J)=DELTANGLE*FEED2(I,J)
690 REM
720 XWORK(I,J)=X(I,J)+0.5*K2(I,J)
725 XFWORK(I,J)=XFEEDBACK(I,J)+0.5*K2FB(I,J)
730 NEXT J
740 NEXT I
750 REM
760 REM THIRD RUNGA-KUTTA CONSTANT
770 REM
780 CALL FCNN(ANGLE+0.5*DELTANGLE,XWORK(,),XFWORK(,),JJ(,),F3(,),FEED3(,),BE,MU,
GAMMA,PSQUARE)
790 REM
800 FOR I=1 TO ORDER
810 FOR J=1 TO ORDER
820 K3(I,J)=DELTANGLE*F3(I,J)
825 K3FB(I,J)=DELTANGLE*FEED3(I,J)
850 REM
```

```
880 XWORK(I,J)=X(I,J)+K3(I,J)
885 XFWORK(I,J)=XFEEDBACK(I,J)+K3FB(I,J)
890 NEXT J
900 NEXT I
910 REM
920 REM FOURTH RUNGA-KUTTA CONSTANT
930 REM
940 CALL FCNN(ANGLE+DELTANGLE,XWORK(,),XFWORK(,),JJ(,),F4(,),FEED4(,),BE,MU,GAMM
A,PSQUARE)
950 REM
960 FOR I=1 TO ORDER
970 FOR J=1 TO ORDER
980 K4(I,J)=DELTANGLE*F4(I,J)
985 K4FB(I,J)=DELTANGLE*FEED4(I,J)
990 NEXT J
1000 NEXT I
1010 REM
1020 REM EVALUATION OF STATE VARIABLES
1030 REM
1040 FOR I=1 TO ORDER
1050 FOR J=1 TO ORDER
1060 XNEW(I,J)=X(I,J)+(1/6)*(K1(I,J)+2*K2(I,J)+2*K3(I,J)+K4(I,J))
1065 XFEEDBACK(I,J)=XFEEDBACK(I,J)+(1/6)*(K1FB(I,J)+2*K2FB(I,J)+2*K3FB(I,J)+K4FB
(I,J))
1070 NEXT J
1080 NEXT I
1090 REM
1100 REM UPDATE STATE VARIABLES
1110 REM
1115 REM PRINT #1:"      ANGLE=",(ANGLE+DELTANGLE)*180/(PI)
1116 IF (N/10)=INT(N/10)THEN 1117 ELSE 1120
1117 REM  PRINT #2:ANGLE+DELTANGLE
1120 FOR I=1 TO ORDER
1130 FOR J=1 TO ORDER
1140 X(I,J)=XNEW(I,J)
1145 REM PRINT #1,USING "   F(#,#)=###.#####^^^^":I,J,XNEW(I,J)
1150 NEXT J
1160 NEXT I
1162 IF (N/10)=INT(N/10)THEN 1165 ELSE 1170
1165 REM PRINT #2:X(1,1):: PRINT #2:X(1,2):: PRINT #2:X(2,1):: PRINT #2:X(2,2)
1170 REM
1180 ANGLE=ANGLE+DELTANGLE
1190 REM
1200 NEXT N
1210 REM
1220 REM PRINT OUTPUT
1230 REM
1231 PRINT #1:"-----------------------------------------------------------------
---"
1232 PRINT #1
1233 PRINT #1,USING "  BE=##.##^^^^   MU=##.##^^^^":BE,MU
1234 PRINT #1
```

```
1235 PRINT #1,USING "  GAMMA=##.##^^^^   PSQUARE=##.##^^^^";GAMMA,PSQUARE
1236 PRINT #1
1237 PRINT #1:"FEEDBACK CONTROL STYSTEM"
1238 PRINT #1
1240 FOR I=1 TO ORDER
1250 FOR J=1 TO ORDER
1260 PRINT #1,USING "    X(#,#)=###.#####^^^^":I,J,XFEEDBACK(I,J)
1270 NEXT J
1280 NEXT I
1290 PRINT #1
1300 PRINT #1
1310 PRINT #1
1320 REM
1325 REM NEXT GAMMA
1326 REM
1330 CLOSE #1
1335 REM CLOSE #2
1340 REM
1350 END
```

```
1360 SUB FCNN(T,XWORK(,),XFWORK(,),JJ(,),F(,),XPRIME(,),BE,MU,GAMMA,PSQUARE)
1370 REM
1380 CALL MATRIXA(T,AMAT(,),BE,MU,GAMMA,PSQUARE)
1390 REM
1400 CALL MATRIXMULT(AMAT(,),XWORK(,),2,2,2,FWORK1(,))
1401 REM
1402 CALL MATRIXMULT(XWORK(,),JJ(,),2,2,2,FWORK2(,))
1403 REM
1404 CALL MATRIXADD(FWORK1(,),FWORK2(,),2,2,F(,),-1)
1406 REM
1410 REM FEEDBACK CONTROL SYSTEM DETERMINED
1420 REM
1430 REM CALL INVERSE(2,XWORK(,),FINVERSE(,))
1431 DET=XWORK(1,1)*XWORK(2,2)-XWORK(1,2)*XWORK(2,1)
1432 FINVERSE(1,1)=XWORK(2,2)/DET
1433 FINVERSE(1,2)=-XWORK(1,2)/DET
1434 FINVERSE(2,1)=-XWORK(2,1)/DET
1435 FINVERSE(2,2)=XWORK(1,1)/DET
1440 REM
1450 K=0.109737
1460 B1=0.0
1470 B2=1.0
1480 REM
1490 FEEDBACK(1,1)=K*B1*FINVERSE(2,1)
1500 FEEDBACK(1,2)=K*B1*FINVERSE(2,2)
1510 FEEDBACK(2,1)=K*B2*FINVERSE(2,1)
1520 FEEDBACK(2,2)=K*B2*FINVERSE(2,2)
1530 REM
1540 CALL MATRIXADD(AMAT(,),FEEDBACK(,),2,2,APRIME(,),1)
1550 REM
1560 CALL MATRIXMULT(APRIME(,),XFWORK(,),2,2,2,XPRIME(,))
1570 REM
1580 SUBEND
```

```
2000 SUB MATRIXA(PSI,MATA(,),BE,MU,GAMMA,PSQUARE)
2001 REM
2002 REM ALGORITHM TO CALCULATE STATE TRANSITION MATRIX
2003 REM FOR A PERIODIC FUNCTION
2004 REM IN PARTICULAR-HELICOPTER BLADE, NO FEEDBACK
2005 REM  UPDATED 16 AUG 1984
2006 REM
2007 DEF ARCSIN(ZZ)=ATN(ZZ/SQR(1-ZZ*ZZ))
2008 PSIT=PSI
2010 IF PSIT>=0.0 AND PSIT<PI THEN 2020 ELSE 2012
2012 IF PSIT>=PI AND PSIT<(PI+ARCSIN(BE/MU))THEN 2030 ELSE 2014
2014 IF PSIT>=(2*PI-ARCSIN(BE/MU))AND PSIT<(2*PI)-.000001 THEN 2030 ELSE 2016
2016 IF PSIT>(2*PI)-.000001 THEN PSIT=0.0 :: GO TO 2020 :: ELSE GO TO 2018
2018 GO TO 2040
2020 REM PRINT #1:"2020"
2022 K=(1/3)*BE^3*MU*COS(PSIT)+(1/4)*BE^2*MU^2*SIN(2*PSIT)
2023 REM
2024 C=(1/4)*BE^4+(1/3)*BE^3*MU*SIN(PSIT)
2025 REM
2026 GO TO 2050
2030 REM PRINT #1:"2030"
2031 REM
2032 K=(1/3)*BE^3*MU*COS(PSIT)+(1/4)*BE^2*MU^2*SIN(2*PSIT)+MU^4*(-(1/12)*SIN(2*P
SIT)+(1/24)*SIN(4*PSIT))
2033 REM
2034 C=(1/4)*BE^4+(1/3)*BE^3*MU*SIN(PSIT)+MU^4*((1/16)-(1/12)*COS(2*PSIT)+(1/48)
*COS(4*PSIT))
2035 REM
2036 GO TO 2050
2040 REM PRINT #1:"2040"
2041 REM
2042 K=-(1/3)*BE^3*MU*COS(PSIT)-(1/4)*BE^2*MU^2*SIN(2*PSIT)
2043 REM
2044 C=-(1/4)*BE^4-(1/3)*BE^3*MU*SIN(PSIT)
2045 REM
2050 REM CONSTRUCT MATRIX A
2051 REM
2052 MATA(1,1)=0
2053 MATA(1,2)=1
2054 MATA(2,1)=-(GAMMA/2)*(2*PSQUARE/GAMMA+K)
2055 MATA(2,2)=-(GAMMA*C/2)
2056 REM
2057 REM RETURN TO MAIN PROGRAM
2060 SUBEND
```

```
4000 SUB INPUTT(ROW,COL,WORK(,),IERROR)
4010 REM  INPUT IN A MATRIX DATA
4020 REM
4030 PRINT
4040 INPUT "ENTER #ROWS,#COLS OF MATRIX-":ROW,COL
4050 IF ROW>8 THEN 4040
4060 IF ROW<2 THEN IERROR=1
4080 FOR I=1 TO ROW
4090 PRINT "ROW #",I
4100 FOR J=1 TO COL
4110 PRINT J,"  "
4120 INPUT WORK(I,J)
4130 NEXT J
4150 NEXT I
4160 SUBEND :: REM FROM THE SUBROUTINE
```

```
5000 SUB INVERSE(N2,A(,),B(,))
5001 REM GAUSS-JORDAN MATRIX INVERSION ROUTINE
5010 REM
5011 REM    IDENTIFIERS
5013 REM A       A      COEFFICIENT MATRIX
5014 REM B       B      WORK MATRIX
5015 REM B1      BIG    BIGGEST VALUE
5017 REM D3      DETERM DETERMINANT
5018 REM E1      ERMES  ERROR FLAG
5019 REM H1      HOLD   WORK VARIABLE
5020 REM I2      INDEX  WORK MATRIX
5021 REM I3      IROW   ROW INDEX
5022 REM I4      ICOL   COLUMN INDEX
5023 REM I5      INVRS  PRINT-INVERSE FLAG
5024 REM N2      NCOL   NUMBER OF COLUMNS
5025 REM P1      PIVOT  PIVOT INDEX
5029 REM     END OF IDENTIFIERS
5080 REM
5090 E1=0 :: REM  BECOMES 1 FOR SINGULAR MATRIX
5100 I5=0 :: REM PRINT INVERSE MATRIX IF ZERO
5110 N3=1 :: REM NUMBER OF CONSTANT VECTORS
5120 FOR I=1 TO N2
5130 FOR J=1 TO N2
5140 B(I,J)=A(I,J)
5150 NEXT J
5170 I2(I,3)=0.
5180 NEXT I
5190 D3=1
5200 FOR I=1 TO N2
5210 REM
5220 REM  SEARCH FOR LARGEST PIVOT ELEMENT
5230 REM
5240 B1=0.
5250 FOR J=1 TO N2
5260 IF I2(J,3)=1 THEN 5350
5270 FOR K=1 TO N2
5280 IF I2(K,3)>1 THEN 6120
5290 IF I2(K,3)=1 THEN 5340
5300 IF B1>=ABS(B(J,K))THEN 5340
5310 I3=J
5320 I4=K
5330 B1=ABS(B(J,K))
5340 NEXT K
5350 NEXT J
5360 I2(I4,3)=I2(I4,3)+1
5370 I2(I,1)=I3
5380 I2(I,2)=I4
5390 REM  INTERCHANGE ROWS TO PUT PIVOT ON DIAGONAL
5400 IF I3=I4 THEN 5540
5410 D3=-D3
5420 FOR L=1 TO N2
5430 H1=B(I3,L)
5440 B(I3,L)=B(I4,L)
```

```
5450 B(I4,L)=H1
5460 NEXT L
5530 REM  DIVIDE PIVOT ROW BY PIVOT ELEMENT
5540 P1=B(I4,I4)
5550 D3=D3*P1
5560 B(I4,I4)=1
5570 FOR L=1 TO N2
5580 B(I4,L)=B(I4,L)/P1
5590 NEXT L
5640 REM
5650 REM  REDUCE NONPIVOT ROWS
5660 FOR L1=1 TO N2
5670 IF L1=I4 THEN 5770
5680 T=B(L1,I4)
5690 B(L1,I4)=0
5700 FOR L=1 TO N2
5710 B(L1,L)=B(L1,L)-B(I4,L)*T
5720 NEXT L
5770 NEXT L1
5780 NEXT I
5790 REM
5800 REM  INTERCHANGE COLUMNS
5810 REM
5820 FOR I=1 TO N2
5830 L=N2-I+1
5840 IF I2(L,1)=I2(L,2)THEN 5920
5850 I3=I2(L,1)
5860 I4=I2(L,2)
5870 FOR K=1 TO N2
5880 H1=B(K,I3)
5890 B(K,I3)=B(K,I4)
5900 B(K,I4)=H1
5910 NEXT K
5920 NEXT I
5930 FOR K=1 TO N2
5940 IF I2(K,3)<>1 THEN 6120
5950 NEXT K
5960 E1=0.
6000 GO TO 6110
6010 PRINT
6020 PRINT "   MATRIX INVERSION   "
6030 FOR I=1 TO N2
6040 FOR J=1 TO N2
6050 PRINT USING "##.###^^^^":B(I,J)
6060 NEXT J
6070 PRINT
6080 NEXT I
6085 BREAK
6090 PRINT
6100 PRINT "DETERMINANT =":D3
6110 SUBEND :: REM IF INVERSE IS PRINTED
```

```
7000 SUB MATRIXMULT(AM(,),BM(,),ROWA,COLA,COLB.OUT(,))
7001 REM
7002 REM MATRIX MULTIPLICATION ROUTINE
7003 REM INPUTS    MATRIX "A", MATRIX "B",DIMENSION OF AAND COLUMN DIMENSION OF
B
7004 FOR I=1 TO ROWA
7005 FOR J=1 TO COLA
7006 OUT(I,J)=0.0
7007 NEXT J
7008 NEXT I
7009 REM
7010 FOR I=1 TO ROWA
7020 FOR K=1 TO COLB
7030 FOR J=1 TO COLA
7040 SUM=AM(I,J)*BM(J,K)
7050 OUT(I,K)=OUT(I,K)+SUM
7060 NEXT J
7070 NEXT K
7080 NEXT I
7090 REM
7999 SUBEND :: REM RETURN TO MAIN PROGRAM
```

```
9000 SUB MATRIXADD(AADD(,),BADD(,),ROWA,COLA,OUT(,),SIGNN)
9010 REM
9020 REM MATRIX ADDITION/SUBTRACTION ROUTINE
9030 REM
9040 REM SIGNN: +1-ADDITION,-1-SUBTRACTION
9050 REM
9060 FOR I=1 TO ROWA
9070 FOR J=1 TO COLA
9080 IF SIGNN=-1 THEN 9100
9090 OUT(I,J)=AADD(I,J)+BADD(I,J)
9095 GO TO 9110
9100 OUT(I,J)=AADD(I,J)-BADD(I,J)
9110 NEXT J
9120 NEXT I
9130 REM
9140 SUBEND
```

Bibliography

1. Banerjee, D. and others. "Identification of State Variables and Dynamic Inflow From Rotor Model Dynamics Test," Journal of the American Helicopter Society, 22,2:28-36 (April 1977).

2. Berry, John D. "Wind Tunnel Testing of an Advanced UH-1 Type Rotor System," Journal of the American Helicopter Society,28,2:19-23 (April 1983).

3. Calico, R.A. and G.S. Yeakel. "Active Attitude Control of a Spinning Symmetric Satellite in an Elliptical Orbit," Journal of Guidance and Control,6:315-318 (July 1983).

4. Calico, R.A. and others. "Time Periodic Attitude Control Problems," AAS/AIAA Astrodynamics Specialist Conference, Paper 83-339:1-13. AAS Publications Office, San Diego CA 1983.

5. Chen, S.Y. "Stability of Two-Bladed Aeroelastic Rotors on Flexible Supports," Journal of the American Helicopter Society, 28,1:34-41 (January 1983).

6. Crimi, Peter. A Method for Analyzing the Aeroelastic Stability of a Helicopter Rotor in Forward Flight, NASA CR-1332, Washington D.C.,1969.

7. Friedmann, Peretz P. "Formulation and Solution of Rotary Wing Aeroelastic Stability and Response Problems," Vertica, 7,2:101-141 (1983).

8. Friedmann, P.P. and Kottapalli, S.B.R. "Coupled Flap-Lag Torsional Dynamics of Hingeless Rotor Blades in Forward Flight," Journal of the American Helicopter Society, 27:28-36 (October 1982).

9. Friedmann, P. and C.E. Hammond. "Efficient Numerical Treatment of Periodic Systems with Application to Stability Problems," International Journal for Numerical Methods in Engineering, 11: 1117-1136 (1977).

10. Friedmann, P. and J. Shamie. "Aeroelastic Stability of Trimmed Helicopter Blades in Forward Flight," Vertica, 1: 189-211 (1977).

11. Friedmann, P. and L.J. Silverthorn. "Aeroelastic Stability of Coupled Flap-Lag Motion of Hingeless Helicopter Blades at Arbitrary Advance Ratios," Journal of Sound and Vibration, 39,4: 409-428 (1975).

12. Friedmann, P.P. "Aeroelastic Modelling of Large Wind Turbines," Journal of the American Helicopter Society, 21,4:17-27 (October 1976).

13. Friedmann, P. and L.J. Silverthorn. "Aeroelastic Stability of Periodic Systems with Application to Rotor Blade Flutter," AIAA Journal, 11: 1559-1565 (November 1974).

14. Friedmann, P. and P. Tong. "Non-Linear Flap-Lag Dynamics of Hingeless Helicopter Blades in Hover and in Forward Flight," Journal of Sound and Vibration, 30: 9-31 (1973).

15. Fuh, J.S. and others. "Coupled Flap-Torsional Response of a Rotor Blade in Forward Flight Due to Atmospheric Turbulence Excitations," Journal of the American Helicopter Society, 28,3: 3-12 (July 1983).

16. Gaonkar, G.H. and others. "The Use of Actuator-Disc Dynamic Inflow for Helicopter Flap-Lag Stability," Journal of the American Helicopter Society, 28,3: 79-88 (July 1983).

17. Grad, J. and M.A. Brebner. "Eigenvalues and Eigenvectors of a Real General Matrix," Collective Algorithms From CACM, N.Y.: Association for Computing Machinery, Algorithm 343: 343P1-343P10 (1982).

18. Ham, Norman D. and others. "Helicopter Rotor Lag Damping Augmentation Through Individual Blade Control," Vertica, 7,4: 361-371 (1983).

19. 00041-90034 Rev. D. HP-41C Math Pac. Application Manual. Hewllett-Packard, Corvallis, OR.

20. Hohenemser, Kurt H. and Shenng-Kuang Yin. "Some Application of the Method of Multiblade Coordinates," Journal of the American Helicopter Society, 17,3:3-12 (July 1972).

21. Horvay, Gabriel. "Rotor Blade Flapping Motion," Quarterly Applied Mathematics,5,2: 149-167 (1947).

22. Houbolt, John C. and George W. Brooks. Differential Equations of Motion for Combined Flapwise Bending, Chordwise Bending, and Torsion of Twisted Nonuniform Rotor Blades. NACA Report 1346 (1958).

23. Huber, H. and others. "Transonic Effects on Helicopter Rotor Blades," Vertica, 7: 87-100 (1983).

24. Jenkins, Julian L. A Numerical Method for Studying the Transient Blade Motions of a Rotor with Flapping and Lead Lag Degrees of Freedom. NASA TN D-4195, October 1967.

25. Johnson, Wayne. "Development of a Comprehensive Analysis for Rotorcraft-I. Rotor Model and Wake Analysis," Vertica, 5: 99-129 (1981).

26. Johnson, Wayne. "Development of a Comprehensive Analysis for Rotorcraft-II. Aircraft Model, Solution Procedure and Applications," Vertica,5: 185-216 (1981).

27. Johnson, Wayne. "Application of Unsteady Airfoil Theory to Rotory Wings," Journal of Aircraft,17: 285-286 (April 1980).

28. Johnson, Wayne. Helicopter Theory. New Jersey: Princeton University Press, 1980.

29. Johnson, Raymond L. and Kurt H. Hohenemser. "On the Dynamics of Lifting Rotors with Thrust or Tilting Moment Feedback Controls," Journal of the American Helicopter Society, 15: 42-54 (January 1970).

30. Johnston, Robert A. "Rotor Stability Prediction and Correlation with Model and Full-Scale Tests," Journal of the American Helicopter Society,21,2: 20-30 (April 1976).

31. Kreyszig, Erwin. Advanced Engineering Mathematics (Fifth Edition). New York: John Wiley and Sons, 1983.

32. Miller, Alan R. BASIC Programs for Scientist and Engineers. Berkeley:Sybex, 1981.

33. Ormiston, Robert A. "Investigations of Hingeless Rotor Stability," Vertica,7,2: 143-181 (1983).

34. Peters, David A. and others. "Calculations of Trim Settings for a Helicopter Rotor by an Optimized Automatic Controller," Journal of Guidance and Control,7: 85-91 (January 1984).

35. Peters, David A. "Flap_Lag Stability of Helicopter Rotor Blades in Forward Flight," Journal of the American Helicopter Society, 20,4: 2-13 (October 1975).

36. Peters, David A. and Kurt H. Hohenemser. "Application of the Floquet Transition Matrix to Problems of Lifting Rotor Stability," Journal of the American Helicopter Society, 16: 25-33 (1971).

37. Prussing, J.E. and Y.K. Lin. "A Closed-Form Analysis of Rotor Blade Flap-Lag Stability in Hover and Low-Speed Forward Flight in Turbulent Flow," Journal of the American Helicopter Society,28,3: 42-46 (July 1983).

38. Rosen, A. and Z. Beigelman. "A Simplified Model of the Influence of Blade Elastic Pitch Variations on Helicopter Rotor Flapping Dynamics," Vertica,7,4:335-360 (1983).

39. Scheid, Francis. Schaum's Outline Series-Theory and Problems of Numerical Analysis. New York: McGraw-Hill Book Company, 1968.

40. Schrage, Daniel P. and David A. Peters. "Effect of Structural Coupling Parameters on the Flap-Lag Forced Response of a Rotor Blade in Forward Flight using Floquet Theory," Vertica,3,5: 177-185 (1979).

41. Sissingh, G.J. "Dynamics of Rotors Operating at High Advance Ratios," Journal of the American Helicopter Society, 13: 56-63 (July 1968).

42. Smith, Jon M. Scientific Analysis on the Pocket Calculator (Second Edition). New York: John Wiley and Sons, 1977.

43. Von Mises, Richard. Theory of Flight. New York: Dover Publishings, Inc., 1959.

44. Wiesel, W. and William Shelton. "Modal Control of an Unstable Periodic Orbit," Journal of Astronautical Sciences, 31: 63-76 (January 1983).

45. Wood,K.D. Aerospace Vehicle Design Volume I Aircraft Design (Third Edition).Boulder: Johnson Publishing Company, 1968.

VITA

Captain James K. March was born on 25 August 1952 in Detroit, Michigan. He graduated from high school in Detroit in 1970 and attended the University of Notre Dame from which he received the degree of Bachelor of Science in Aerospace Engineering in May 1974. Upon graduation, he received a commission in the USAF through the ROTC program. He was called upon active duty in September 1976. His first duty assignment was as a High Energy Laser Weapons Effects Officer for the Deputy of Engineering, Aeronautical Systems Division, Wright-Patterson AFB, Ohio. He was next assigned in June 1980 to the Deputy of Development and Plans, Armament Division, Eglin AFB, Florida, as a Lead Aircraft/Weapons Non-Nuclear Survivability Engineer. He maintained this position until entering the School of Engineering, Air Force Institute of Technology, in May 1983.

Permanent address: 4119 Meadowsweet
Dayton, Ohio
45424

END

FILMED

6-85

DTIC